The Dissertation Committee for Gang Xu Certifies that this is the approved version of the following dissertation:

Layout Optimization Algorithms for VLSI Design and Manufacturing

Committee:

David Pan, Supervisor

Martin Wong, Supervisor

Al Mok

Gordon Novak

Warren Grobman

Layout Optimization Algorithms for VLSI Design and Manufacturing

by

Gang Xu, BS, MS

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2007

Dedication

To my families.

Acknowledgements

First of all, I would like to thank my supervisors, Prof. Martin D. F. Wong and Prof. David Pan for their constant patience, endless support, and inspirational guidance during my graduate study at UT-Austin. The experience of working under their supervision is invaluable. I will always cherish what I have learned from them in my life.

I also highly appreciate my committee members, Prof. Aloysius K. Mok, Prof. Gordon Novak, and Dr. Warren Grobman for their invaluable comments and suggestions on my research work. Their insights are of great help with the completion of this dissertation.

I am greatly indebted to Dr. Ruiqi Tian and Dr. Li-Da Huang for the collaboration. I also would like to thank all of my colleagues from Prof. Wong's VLSI CAD group and Prof. Pan's UTDA group for those interesting and stimulating discussions: Dr. Xiaoping Tang, Dr. Hung-Ming Chen, Dr. Muzhou Shao, Dr. Hua Xiang, Dr. Haoxing Ren, Mr. Tao Luo, Mr. Peng Yu, Mr. Xiaokang Shi, and others.

Finally, I would like to express my gratitude to my families for their love and support for years. I dedicate this work to them.

Layout Optimization Algorithms for VLSI Design and Manufacturing

Publication No._____

Gang Xu, Ph.D.

The University of Texas at Austin, 2007

Supervisors: David Z. Pan and Martin D.F. Wong

As the feature size of the transistor shrinks into nanometer scale, it becomes a grand challenge for semiconductor manufacturers to achieve good manufacturability of integrated circuits cost-effectively. In this dissertation, we aim at layout optimization algorithms from both manufacturing and design perspectives to address problems in this grand challenge. Our work covers three topics in this research area: a redundant via enhanced maze routing algorithm for yield improvement, a shuttle mask floorplanner, and optimization of post-CMP topography variation.

Existing methods for redundant via insertion are all post-layout optimizations that insert redundant vias after detailed routing. In the first part of this dissertation, we propose the first routing algorithm that conducts redundant via insertion during detailed routing. Our routing problem is formulated as a maze routing with redundant via constraints and transformed into a multiple constraint shortest path problem, and then solved by Lagrangian relaxation technique. Experimental results show that our algorithm

can find routing solutions with remarkably higher rate of redundant via insertion than conventional maze routing.

Shuttle mask is an economical method to share the soaring mask cost by placing different chips on the same mask. Shuttle mask floorplanning is a key step to pack these chips according to certain objectives and constraints related to mask manufacturing and cost. In the second part of this dissertation, we develop a simulated annealing based floorplanner that can optimize these objectives and meet the constraints simultaneously.

Chemical-mechanical polishing (CMP) is a crucial manufacturing step to planarize wafer surface. Minimum post-CMP topography variation is preferred to control the defocus in lithography process. In the third of this dissertation, we present several studies on optimization of the variation. First, we enhance the shuttle mask floorplanner to minimize the post-CMP topography variation. Then we study the following single-block positioning problem: given a shuttle mask floorplan, how to determine a movable block's optimal position with respect to post-CMP topography variation. We propose a fast incremental algorithm achieving 6x to 9x speedup. Finally, we formulate a novel CMP dummy fill problem that targets at minimizing the height variance, which is key to reduce the image distortion by defocus. Experimental results show that with the new formulation, we can significantly reduce the height variance without sacrificing the height spread much.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 IC Manufacturing: The Grand Challenge in Nanometer Era

Ever since the innovative invention of the integrated circuit (IC) by Jack Kilby in 1958 [1], the past decades have witnessed how these small silicon chips gradually prevail and play an indispensable role in our life. Nowadays IC's are omnipresent. They can be found in almost every device, from microwaves to hearing aids, from automobiles to space shuttles, and from Wii to iPhone. It is hard to imagine what the modern human society would look like without the existence of IC's.

The development of the IC manufacturing process never ceases to meet increasing demands for new products with higher performance and stronger functionality. The IC manufacturing process has been evolving generation by generation, each of which refers to as a technology node, and represented by the feature size of the transistor [2]. The feature size of the transistor is measured by half of the distance between two memory elements in a dynamic random access memory (DRAM). As the feature size scales down continuously, a single transistor runs faster and consumes less power, which enables IC designers to pack more transistors into a single chip and make the chip more powerful. The trend of increasing number of transistors follows Moore's law, which predicts that the number of transistors being packed into a single integrated circuit doubles around every 18 months [3]. For example, Intel's 80386 CPU, released in 1985, had 275 thousand transistors at the technology node of 1.5 micron, while in 2006, the semiconductor giant launched Core Duo, a new dual core CPU fabricated with the 65nm technology, which enables the placement of 151 million transistors on the chip.

However, as the feature size of the transistor shrinks into nanometer scale, it becomes a grand challenge for IC manufacturers to achieve both good manufacturability and cost efficiency. On one hand, IC manufacturers have to face up to many new difficulties and complications emerging in the more and more sophisticated manufacturing process, such as increasing circuit complexity, sub-wavelength lithography, and use of new materials for interconnect and dielectric. On the other hand, new techniques solving these difficulties inevitably pump up the cost. Cooperative efforts from both manufacturers and designers are required to conquer the new challenge.

1.2 A CASE STUDY: SUB-WAVELENGTH LITHOGRAPHY

Sub-wavelength lithography [4] is a good example to illustrate the necessity of cooperation between manufacturers and designers. Optical lithography is a major step in the IC manufacturing process. In this step, the circuit layout designs are printed onto the wafer through a mask set and a lens system, as shown in Figure 1.1. Afterwards, the printed image on the wafer will be the basis to form the real circuit layout on the silicon. Therefore, to ensure that the functionality and the performance of the circuit meet the design target, the printed image must keep high fidelity with the original design. Two dominant factors affecting the fidelity are the pitch resolution and feature resolution of optical lithography, which were improved by reducing the light wavelength of the optical lithography equipment in the past [5].

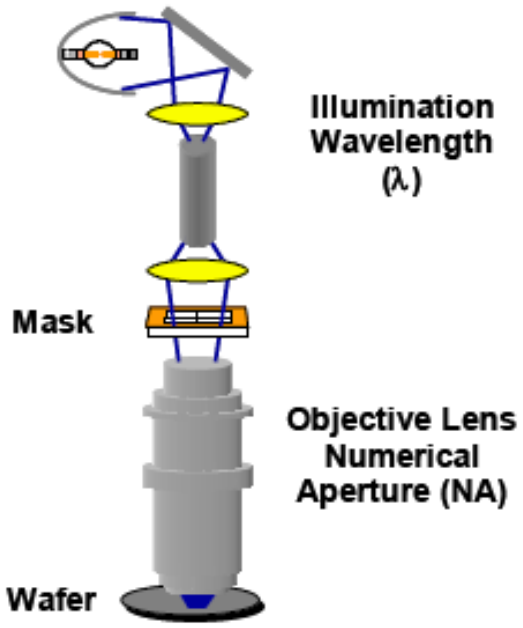


Figure 1.1: Optical lithography for IC manufacturing from [5].

Unfortunately, over the past decades, the development of optical lithography equipment has not been as rapid as scaling down of the IC feature size. As shown in Figure 1.2, starting from 180nm technology node, the wavelength of the light of the state-of-art optical lithography equipment falls behind with the feature size of the IC being fabricated. Consequently, light diffraction starts to affect the resolution, and the problem of image distortions arises. This challenge is called sub-wavelength lithography gap [6]. Notice that in Figure 1.2, IC's at the 90nm technology node were expected to be fabricated by the optical lithography equipment with the wavelength of 157nm, while in reality manufacturers are still using the system of 193nm because of the immaturity of the next generation lithography. In fact, this gap is actually getting wider in time.

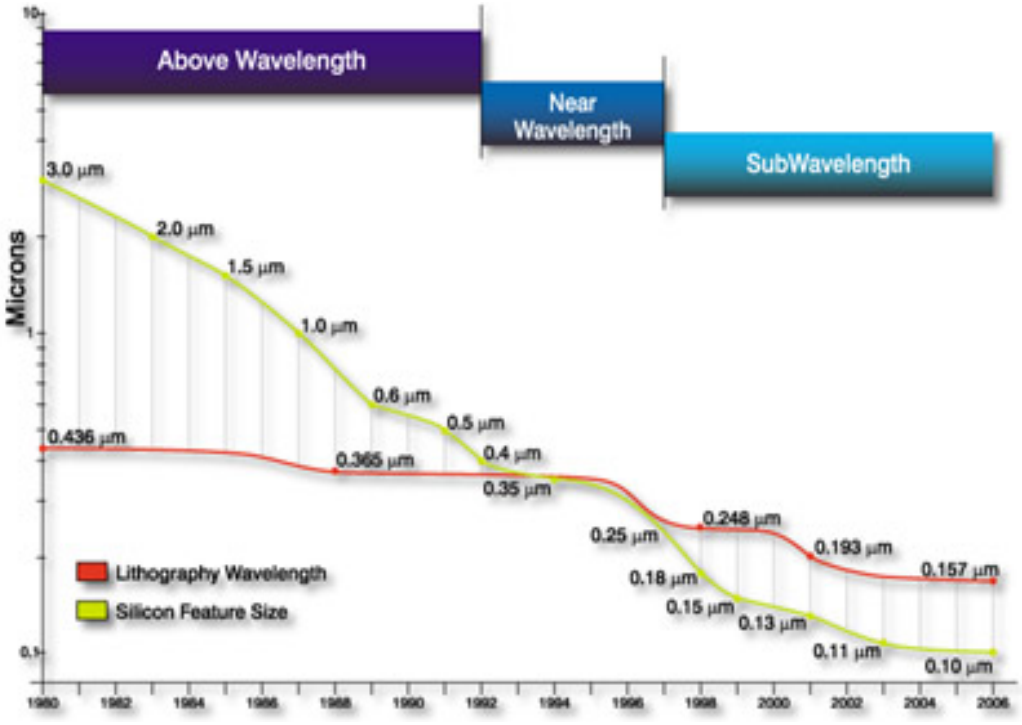


Figure 1.2: Sub-wavelength lithography gap from Synopsys [6].

In order to correct the image distortion, the optical lithography has to extensively use complicated advanced resolution enhancement technologies (RET's) [7], such as optical proximity correction (OPC) [8, 9, 10], phase shifting mask (PSM) [11, 12, 13, 14], and off axis illumination (OAI) [15].

Figure 1.3 illustrates the example of OPC from [16]. Let sub-wavelength lithography be a function *F*. Given an original design layout *X*, the distorted image is *F(X)*. *F(X)* is not identical to *X* because of the image distortion. The essential idea of OPC technique is to find another layout *Y*, such that *F(Y)* is equal to *X*, or at least the error

F(Y) - *X* is within tolerance. The layout *Y*, known as post-OPC layout, will be used to make the mask set.

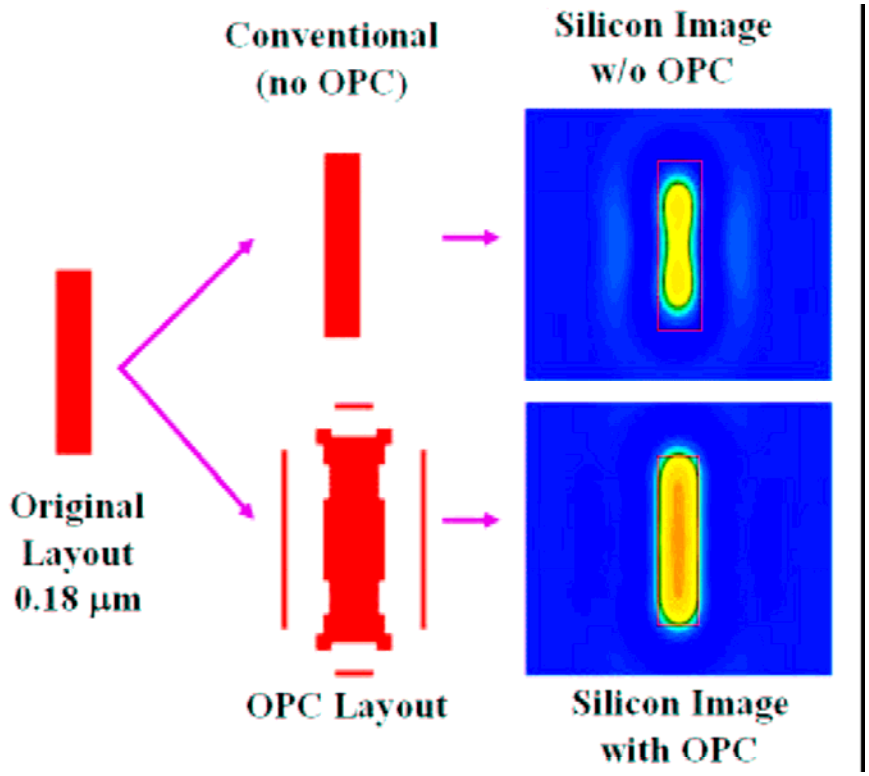


Figure 1.3: The example of OPC from [16].

Although the idea of OPC seems natural and simple, in practice it is computationally very expensive. In addition, as the above OPC example indicates, the post-OPC layout becomes much more complicated, which dramatically pushes up the cost of mask set in optical lithography process, because the volume of data written to the mask and data complexity are quickly increasing.

To solve these problems, manufacturers and designers are striking the target with different strategies. From the manufacturer's side, efforts are made to accelerate the OPC

computation, either by using simplified rule-based OPC to relax the fidelity, or by developing fast OPC algorithms and building dedicated hardware [17]. Efforts are also made to compress the data volume by using new format of layout representation [18] and to save the mask cost by sharing the mask set among different designs [19]. On the other hand, designers also make great contributions to improve OPC process by providing an "OPC friendly" original design layout, which facilitates the computation of the post-OPC layout and reduce the complexity of the post-OPC layout [16, 20].

1.3 MOTIVATION AND CONTRIBUTIONS

The previous case study on sub-wavelength lithography has already revealed the motivation of our research work demonstrated in this dissertation: we aim at layout optimization algorithms from not only manufacturing but also design perspectives to address problems in the grand challenge of IC manufacturing. Our work covers three topics in this research area: a redundant via enhanced maze routing algorithm for yield improvement, a shuttle mask floorplanner, and optimization of post-CMP topography variation.

The redundant via enhanced maze routing algorithm is designed for yield improvement. Redundant via refers to the backup via in addition to the original via in design layout. Redundant via insertion is highly recommended by major foundries to improve yield by reducing via failure [21]. However, existing methods are all post-layout optimizations that insert redundant via after detailed routing. We propose the first routing algorithm that conducts redundant via insertion during detailed routing. Our routing problem is formulated as a maze routing with redundant via constraints. We propose an edge cost function that transforms the problem into a multiple constraint shortest path

problem. The problem is then solved by Lagrangian relaxation technique. Experimental results show that our algorithm can find routing solutions with remarkably higher rate of redundant via insertion than conventional maze routing.

The shuttle mask floorplanner targets at optimization on mask cost and manufacturability. As mentioned earlier in the case study on sub-wavelength lithography, nowadays the mask costs are soaring because of the extensive use of RET's. For example, the mask cost may easily reach one million dollars at 130nm technology node and two million at 90nm node. Particularly, for a low product volume design, e.g., an ASIC prototype, such a high cost is unfavorable, and sometimes even unaffordable, because it is impossible to amortize the cost over the product volume. Shuttle mask is an economical method to share the mask cost by putting different chips on the same mask. Shuttle mask floorplanning is a key step to pack these chips according to certain objectives and constraints related to mask manufacturing and cost, including area minimization, maximization of wafer utilization, and die-to-die inspection constraint. We develop a simulated annealing based floorplanner that can optimize these objectives and meet the constraints simultaneously.

Chemical-mechanical polishing (CMP) is a crucial manufacturing step to planarize wafer surface. The minimum post-CMP variation is preferred to control the defocus in lithography process. In this dissertation, we present several studies on optimization of post-CMP topography variation. Based on an analytical model that uses a 2-D low pass filter to calculate the post-CMP topography variation of inter-layer oxide (ILD), we enhance the shuttle mask floorplanner in Chapter 3 to be "CMP-aware", that is, the floorplanner aims at minimizing the post-CMP topography variation.

We also notice a new problem in CMP-aware shuttle mask floorplanning. Given a slicing (or grid) shuttle mask floorplan, some block might be movable within its enclosing rectangle. The problem of determining the movable block's optimal position with respect to post-CMP topography variation arises. We formulate the problem as a single-block positioning problem (SBPP). By applying the linear and the shift property of the convolution to the incremental layout, our algorithm replaces the *O(nlogn)* FFT operation with a simple *O(n)* matrix addition in loop iteration, and thus runs much faster. The experimental results show 6x to 9x speedup consistently compared with the non-incremental counterpart.

In the last study, we present a novel CMP dummy fill problem formulation. The CMP dummy fill problem seeks the optimal scheme of dummy feature fill with respect to the minimum post-CMP variation. The traditional formulation tries to reduce the height spread of the layout, which is a linear objective. Instead, our new formulation targets at minimizing the height variance, a quadratic objective. In our opinion, this objective is more important to reduce the total image distortion by defcus. Experimental results show that with the new formulation, we can significantly reduce the height variance without sacrificing the height spread much.

The rest of this dissertation is organized as follows. Chapter 2 discusses the redundant via enhanced maze routing. Chapter 3 presents the multi-objective shuttle mask floorplanner. Chapter 4 focuses on those studies on CMP optimization. Finally, Chapter 5 concludes the dissertation and discusses the future work.

Chapter 2: Redundant Via Enhanced Maze Routing for Yield Improvement

2.1 INTRODUCTION

As we mentioned earlier in Chapter 1, when the feature size continues to shrink to nanometer regime, IC designers' participation has been called for the grand challenge of IC manufacturing. At the new technology nodes, designers have to consider manufacturability and yield related problems in the design flow in order to help relieve the heavy burden carried by manufacturers. How to efficiently solve these new problems forms an active EDA research topic, known as "design for manufacturability" (DFM) [22, 23, 24].

Among DFM problems, how to reduce yield loss by via failure is one of the most important. Vias are components in VLSI circuits to connect wire segments on different metal layers. Vias have high resistance, which suggests an important influence on RC delay. For example, in TSMC 180nm technology, the resistance of two via stacks at each end of M1 wire will be around 20 ohm, equivalent to about 0.1 mm wire [25]. Therefore, vias have significant impact on both functionality and performance.

However, vias may fail partially or completely due to various reasons such as misalignment, electromigration, and thermal stress induced voiding, which become more severe in nanometer era [26, 27, 28, 29]. A complete via failure may lead to a broken net and result in the mistaken functionality. A partial via failure will increase the resistance, bring out unexpected RC delay, and result in timing problems. Nowadays, circuits at new

technology nodes have more vias in the layout, as the circuit complexity increases, feature orientation becomes stricter [30], and extra vias are introduced by jumper insertion to fix the antenna effect [31]. Yield loss by via failure thus becomes more critical and requires a careful control.

A nice solution to reduce yield loss by via failure is to add a redundant via adjacent to each single normal via as a backup, as shown in Figure 2.1. Here we refer to the single normal via as a via on the wire with minimal wire width. In the rest of this chapter, a via usually refers to a single normal via. Redundant vias can greatly reduce the likelihood of broken nets. Assume a via has 10% probability to completely fail. Consider a simple model in which the redundant via fails at the same probability independently. The probability that the net is broken because of this via failure will become 1%, which is much less. As the matter of fact, data in [32] has shown that by adding redundant vias, the via yield can be maintained on a stable level even when the misalignment issue becomes more serious. Besides, redundant vias also alleviate the delay penalty by partial via failures, because the redundant via can serve as a conductor parallel to the original via and decrease the resistance.

Because of its benefits in reducing via failure, redundant via insertion has been strongly recommended by major foundries in their 130nm and 90nm processes [21]. Meanwhile, major EDA vendors such as Cadence and Synopsys have already added the feature of redundant via insertion to their latest routers (Cadence Nanoroute, Synopsys Astro). There are also third-party EDA tools such as Nannor Acuma and Prediction EYE/PEYE [33] specially designed to insert redundant vias.

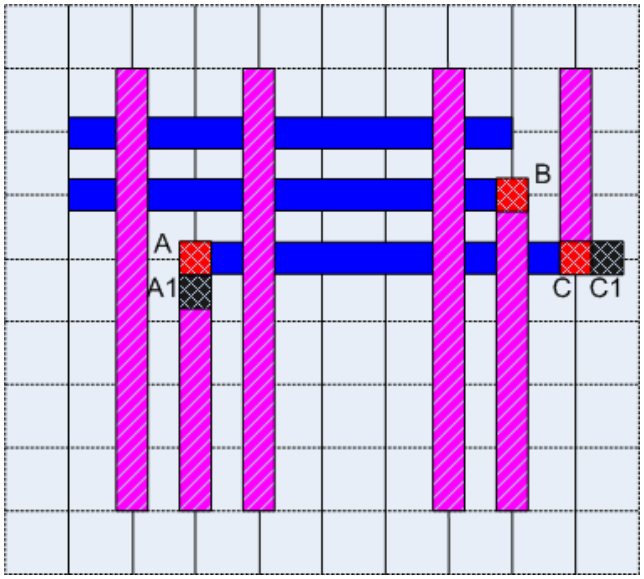


Figure 2.1: Redundant vias. *A1* and *C1* are redundant vias of *A* and *C* respectively. We are unable to insert the redundant via for *B* because of the minimum spacing rule.

However, all these tools are conducting redundant via insertion in the post-layout stage following detailed routing. That is, redundant vias are inserted after the layout is almost determined. Because at this stage only slight layout modifications are allowed, this methodology will inevitably restrict the feasibility of redundant via insertion. A better idea is to consider the redundant via insertion in the routing stage, which has been foreseen as one of the future routing challenges in nanometer era [34].

In this chapter, we propose a maze routing algorithm that considers the feasibility of redundant via insertion in the detailed routing stage. To our best knowledge, this is the first study in this direction in public domain. In our algorithm, 2-pin nets are routed with the constraint on the maximum number of dead vias in each net to reflect the redundant via insertion in the future. Here dead vias refer to vias ineligible to have redundant vias.

For example, via B in Figure 2.1 is a dead via. We propose an edge cost function. Based on the cost function, the maze routing problem with redundant via constraints is transformed to a multi-constraint shortest path problem, and solved by Lagrangian relaxation technique. Experimental results show that our algorithm can find routing layout with much higher rate of redundant via insertion than conventional maze routing.

The rest of this chapter is organized as follows. Section 2.2 presents the problem formulation. Section 2.3 studies the solution to a special case, and then extends it to the general case. Experimental results are shown in section 2.4. Section 2.5 will conclude the chapter.

2.2 PROBLEM FORMULATION

In this chapter, we use the maze routing algorithm, which is a grid based sequential routing algorithm. The routing region in maze routing is represented as a k-layer grid graph. An x-axis or y-axis edge on a layer represents a wire segment. A via corresponds to a z-axis edge connecting a pair of vertices at the same x-y coordinate on the two neighboring layers. Obstacles and occupied vertices and edges are removed from the graph because they are not available as routing resource. In the following part of this chapter we refer the vertex to a via for simplicity. As a sequential routing algorithm, maze routing seeks net routes one by one in a certain pre-set order. Once a net is routed, the vertices and edges representing pins, vias, and wire segments of this net are occupied and then removed. Nets being routed late will have less routing resource. See [35] for more details about maze routing.

Some basic concepts are required to be introduced first for discussion of

redundant via constraint. For any vertex *v* representing a single normal via in the grid graph, we define its adjacent vertices as neighbors. The unoccupied neighbors of *v* are called *off-track neighbors*, and the neighbors only occupied by the net that *v* belongs to are called *on-track neighbors*. On-track and off-track neighbors of v are *free neighbors*. The total number of free neighbors of v is defined as the *degree of freedom (DoF)* of *v*. Vias with non-zero DoF are *alive*. Otherwise, they are *dead*. A via with only one free neighbor is *critical*. Figure 2.2 shows examples of these concepts.

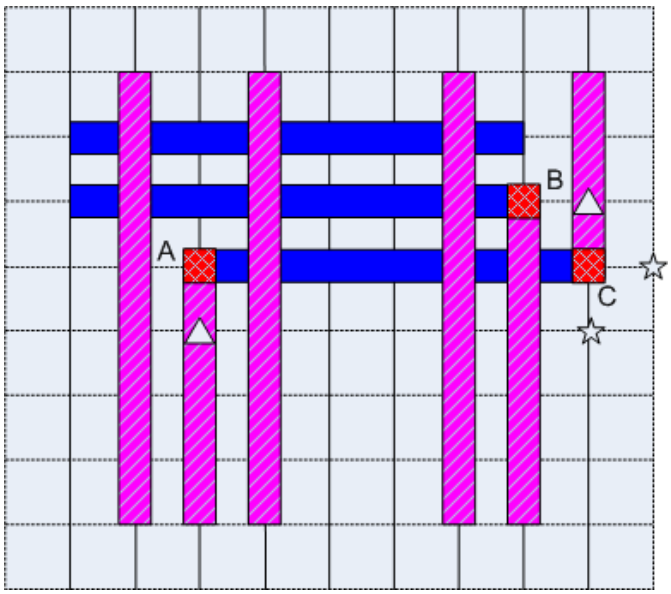


Figure 2.2: Free neighbors and the degree of freedom of a via. Stars and triangles indicate free neighbors of *A* and *C*. Stars are off-track neighbors; triangles are on-track neighbors. *A* is a critical via because its DoF is 1. *B* is a dead via. The DoF of *C* is 3.

The redundant via must be inserted between *v* and one of its free neighbors in order to satisfy the minimum spacing rule, as shown in Figure 1. Therefore, only live vias

can have redundant vias. In our problem formulation, we constrain the maximum number of dead vias in each net to guarantee redundant via insertion. The constraint is per net in order to control criticality of different nets. For example, consider some nets that are not timing critical. As we discussed in Section I, partial via failure may lead to increasing resistance and timing loss. However, because of the non-criticality, timing loss of these nets caused by the partial via failure may be acceptable. These net can have a higher budget of dead vias.

In addition, we take a strategy called *delayed insertion* to insert redundant vias. That is, redundant vias are not inserted until all nets are routed. During the routing stage, we just keep track of free neighbors of each via. The advantage of the delayed insertion is that the router is allowed to kill live vias in the routed nets to get a better route for the new net, as long as the constraints of dead vias are still satisfied. Here a via is *killed* if all of its free neighbors are occupied by the new net. Figure 2.3 shows such an example.

Assuming net *m* is routed, the maze routing problem with redundant via constraints is formulated as follows:

Problem 1: Maze routing with redundant via constraints (MRRVC):

Find the shortest route for net m such that $\forall i::1 \le i \le m : DV_i \le C_i$ *, where* DV_i *is the number of dead vias in net i,* C_i *is the constraint of net i.*

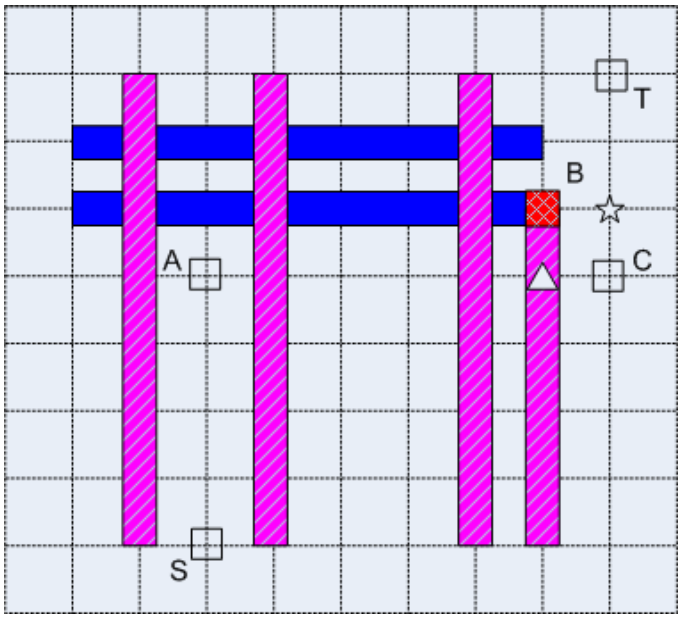


Figure 2.3: Delayed insertion. Compare this layout to the one in Figure 2. In this layout *B* has two free neighbors marked by triangle and star. After routing a new net from S to T passing *A* and *C*, as shown in Figure 2, *B* is killed.

2.3 PROBLEM SOLUTION

In this section, we present the solution to the MRRVC problem. First, we study a special case of this problem. The problem is transformed to a multi-constraint shortest path problem, and solved by Lagrangian relaxation technique. Then we extend this solution to the general case.

2.3.1 The solution to MRRVC: a special case

The following special case is considered in this section: before the m-th net is routed, all live vias are critical. In this scenario, the routing layout is dense and every live via is easy to be killed, as shown in Figure 2.4.

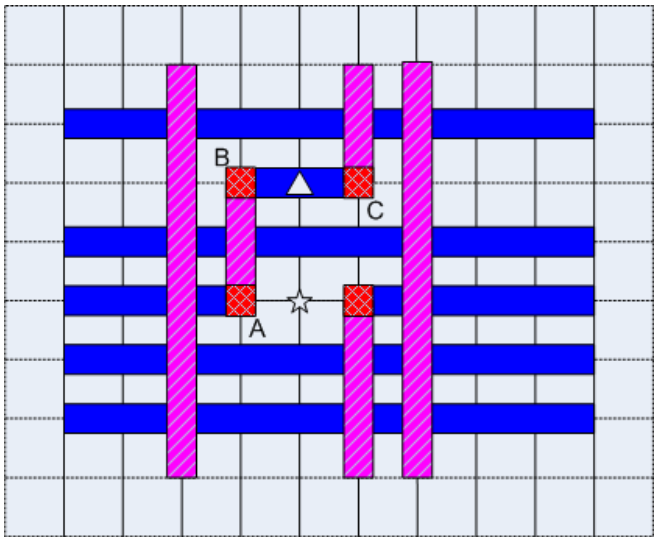


Figure 2.4: A special case that all live vias are critical.

In brief, the flow of the solution is as follows. First, the MRRVC problem is mapped to an equivalent multi-constrained shortest path problem (MCSP) by assigning each edge a cost vector. Second, based on the MCSP problem, a Lagrangian sub-problem (LSP) and a Lagrangian multiplier problem (LMP) are constructed. Because the optimal solution to LMP is proved the lower bound of the solution to its corresponding MCSP problem, the search procedure of LMP solution can serve as a heuristic to find the solution of MCSP, which is also a solution to MRRVC. In the rest of this section details of this flow will be discussed.

The first step is to assign cost vector to every edge in the current routing graph. By doing this we can count how many vias in the routed net *i* will be killed when routing a new net *m*, and map the MRRVC problem into a MCSP problem. The algorithm of cost assignment is shown in Figure 2.5. Initially, costs of all edges are set to zero. Then

algorithm scans free neighbors of each live via, and increases cost of each incident edge to the free neighbor by one. By applying the above algorithm to each routed net *i, i=1 to m-1,* the edge in the current routing graph is assigned a cost vector $(c_1^e, c_2^e, ..., c_{m-1}^e)$.

```
Input: net i, the routing graph
Output: cost assignment

for each edge e in the routing graph
    cost(e) = 0;
for each live via v in i
     for each free neighbor n of v
          for each incident edge e to n
               cost(e) ++;
end.
```

Figure 2.5: The algorithm of edge cost assignment.

Now we have the following theorem to count killed vias.

Theorem 1: $\forall i :: 1 \le i \le m-1 : \sum_{e \in m} c_i^e = KV_i$, *where e represents an edge,* c_i^e *is cost of e regarding net i,* KV_i *is the total number of vias in net i that are killed by net m if net m is routed.*

Proof: A via *v* in net *i* is killed by net *m* if and only if its free neighbor is occupied by the new net *m,* because *v* is critical. On the other hand, consider a vertex *n*, which is the free neighbor of a via *v* in net *i*. If *n* is occupied by net *m* that forms a path, *n* must be an internal node of *m* and there is exactly one in edge of *n*, expressed as *e*,

occupied by *m*. According to the algorithm of edge cost assignment, c_i^e indicates the number of vias in net *i* that need *n* as the free neighbor for redundant via insertion. These vias will be killed by net m, as *n* is occupied. For any edge e' that is not an in edges of a free neighbor of net *i*, the edge cost $c_i^{e'}$ is 0 according to the algorithm of edge cost assignment. Therefore, summing up edge cost c_i^e along the net *m* will get the number of vias in net *i* that are killed by net *m*.

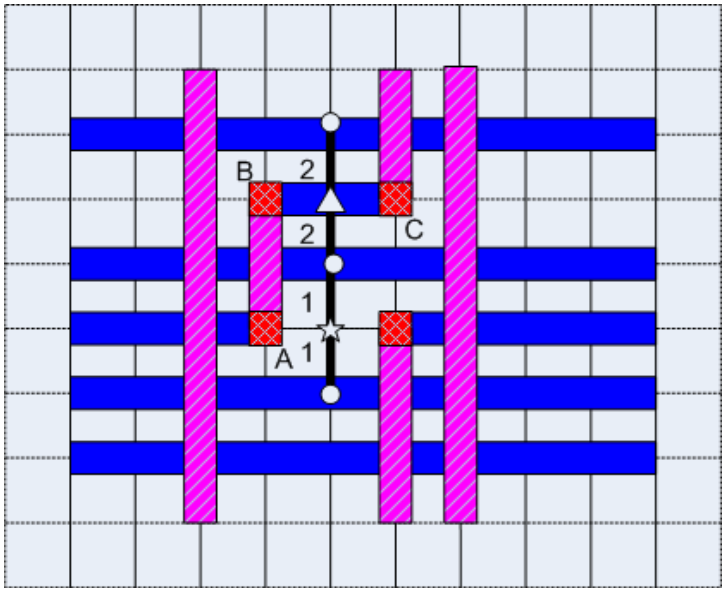


Figure 2.6: The edge cost assignment after the algorithm terminates. Here net *i* refers to the net that vias *A*, *B*, and *C* belong to. For simplicity, we only draw edges with non-zero cost. The direction of the edge is also ignored. Costs of in edges of the free neighbor marked by triangle are both 2, because the triangle is a free neighbor of both *B* and *C*. Costs of in edges of the star vertex are both 1, because one neighboring via of its is not in net *i*.

The result can be verified in Figure 2.6. Any new net passing the star and the triangle, which represent the free neighbors of net containing critical vias *A*, *B*, and *C*,

will lead to sum of edge cost to increase by 3, regardless of the direction. The new net will kill 3 vias *A, B* and *C* that are in the same net.

Notice that routed nets can also kill vias in the new net. It is easy to count these vias. We just assign cost to each z-axis edge in the current graph in the following way: looking on the z-axis edge as a via, assign one if it is dead, and zero otherwise. Assign zero to all x-axis and y-axis edges. The edge cost is denoted as c_m^e. An example of this cost assignment is shown in Figure 2.7 and Figure 2.8. We have the following theorem to count dead vias in the new net *m*.

Theorem 2: $\sum_{e \in m} c_m^e = DV_m$, *where e represents an edge,* c_m^e *is cost of e regarding net m,* DV_m *is the total number of dead vias of net m.*

Proof: Consider any dead via in net *m*. Its neighbors must have been occupied before routing net *m*, from the definition of free neighbor. So its edge cost is one. Costs of all other edges are zero. Q.E.D.

To count dead vias in each net after net i is routed, we assign a cost vector $(c_1^e, c_2^e, ..., c_{m-1}^e, c_m^e)$ to each edge *e* in the current graph, where $c_1^e, c_2^e, ..., c_{m-1}^e$ are assigned in the way of theorem 1, and c_m^e is assigned based on theorem 2.

Based on the cost assignment, the original MRRVC is transformed into the following MCSP problem.

Problem 2: Multi-constrained shorted path (MCSP)

Given a graph where each edge is assigned a cost vector $(c_1^e, c_2^e, ..., c_{m-1}^e, c_m^e)$, *two vertices s and t in the graph, and a constraint vector* $(C'_1, C'_2, ..., C'_{m-1}, C'_m)$, *also assume in any net i, i<m, there exists* DV'_i *dead vias already, find a shortest path P from s to t such that* $\forall i :: 1 \le i \le m : \sum_{e \in P} c_i^e \le C'_i$, *where* $C'_i = KV_i = C_i - DV'_i$ *if i<m, and* $C'_m = C_m$.

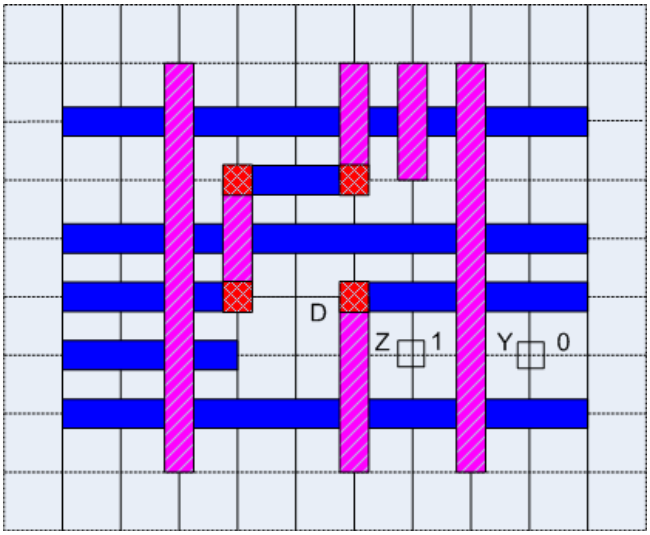


Figure 2.7: The edge cost assignment to the z-axis edge. The two squares indicate two z-axis edges *Y* and *Z*. If *Z* were a via in the new net, it would be a dead via. Therefore, its cost is 1. *Y* would have a free neighbor. Its cost will be 0.

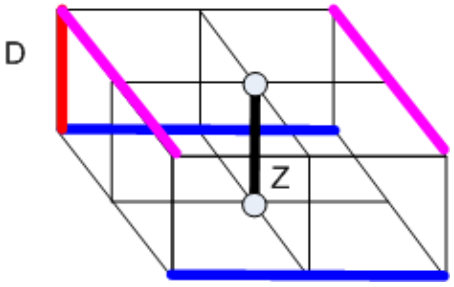


Figure 2.8: A 3-D view of the local part of Figure 2.7 about *Z*

The MCSP problem was studied in [36] and [16]. It is proved a NP-hard problem by reducing it to a well-known NP-complete problem: 3-partition problem. Because of the NP-hardness, we propose a heuristic solution based on Lagrangian relaxation technique. The solution starts with the construction of the Lagrangian sub-problem (LSP) and the Lagrangian multiplier problem (LMP).

Given a MCSP problem instance and a non-negative constant vector $(\lambda_1, \lambda_2, ..., \lambda_{m-1}, \lambda_m)$, in which each element represents the weight of a constraint, we can construct the following unconstrained Lagrangian sub-problem.

Problem 3: Lagrangian Sub-Problem (LSP)

Minimize $\sum_{e \in P} 1 + \sum_{i=1}^{m} \lambda_i (\sum_{e \in P} c_i^e - C'_i)$.

Equivalently, it is:

Minimize $\sum_{e \in P} (1 + \sum_{i=1}^{m} \lambda_i c_i^e) - \sum_{i=1}^{m} \lambda_i C'_i$ *, where variable P denotes net m, the net to be routed.*

Because the last term is constant, the above LSP can be solved optimally by the weighted shortest path algorithm in polynomial time [37].

The element in the vector of constraint weight is known as Lagrangian multiplier. Let the multiplier be variable, the Lagrangian multiplier problem (LMP) for MCSP is defined as follows.

Problem 4: Lagrangian Multiplier Problem (LMP).

Maximize $L(\lambda_1, \lambda_2, ..., \lambda_{m-1}, \lambda_m)$

subject to $\forall i :: 1 \le i \le m : \lambda_i \ge 0$.

The nice property of LMP is that the optimal solution is the lower bound of the optimal solution of MCSP because of the following inequality.

For any $(\lambda_1, \lambda_2, ..., \lambda_{m-1}, \lambda_m)$, $\lambda_i \ge 0$,

$$\min_P \{\sum_{e \in P} 1 + \sum_{i=1}^{m} \lambda_i (\sum_{e \in P} c_i^e - C'_i)\} = L(\lambda_1, \lambda_2, ..., \lambda_{m-1}, \lambda_m)$$

$$\le \min_P \{\sum_{e \in P} 1 + \sum_{i=1}^{m} \lambda_i (\sum_{e \in P} c_i^e - C'_i) : \forall i :: 1 \le i \le m : \sum_{e \in P} c_i^e \le C'_i\}$$

$$\le \min_P \{\sum_{e \in P} 1 : \forall i :: 1 \le i \le m : \sum_{e \in P} c_i^e \le C'_i\}$$

Proof: The inequalities can be proved as follows. Inequality $(1) \le (2)$ holds, because (1) is the optimal solution to the LSP while (2) is the solution as the LSP is constrained in the sub-space: $\forall i :: 1 \le i \le m : \sum_{e \in P} c_i^e \le C'_i$. The solution space of the constrained LSP is a subset of the solution space of the unconstrained LSP. Therefore, the optimal solution in the sub-space, i.e., (2), is at most as good as the optimal solution in the whole space, i.e. (1). Inequality $(2) \le (3)$ also holds, as the term $\sum_{i=1}^{m} \lambda_i (\sum_{e \in P} c_i^e - C'_i)$ is non-negative. From the transitivity, we have proved the inequality $(1) \le (3)$. Notice that this inequality holds for any $(\lambda_1, \lambda_2, ..., \lambda_{m-1}, \lambda_m)$, $\lambda_i \ge 0$, including the solution to LMP, the lower bound property mentioned above is proved. Q.E.D.

Therefore, if there exist $(\lambda_1, \lambda_2, ..., \lambda_{m-1}, \lambda_m)$ and path P such that

$L(\lambda_1, \lambda_2, ..., \lambda_{m-1}, \lambda_m) = \sum_{e \in P} 1$ and $\forall i :: 1 \le i \le m : \sum_{e \in P} c_i^e \le C'_i$, P will be the optimal solution to MCSP and $(\lambda_1, \lambda_2, ..., \lambda_{m-1}, \lambda_m)$ is the optimal solution to LMP. In addition, LMP is a convex programming problem, and thus can be solved by non-linear programming techniques, for example, the sub-gradient method [38]. Therefore, the solution to MCSP can be approximate by solving LMP.

The following algorithm shown in Figure 2.9 is a sub-gradient method to solve MCSP. After cost assignment by using the algorithm in Figure 2.5, the LSP problem is solved in each iteration and the Lagrangian multipliers are updated accordingly. The algorithm converges to the optimal solution of LMP if $\lim_{t \to \infty} \theta_t = 0$ and $\lim_{t \to \infty} \sum_{i=1}^{t} \theta_t = \infty$ [38].

cost assignment in Figure 5
for each λ_i $\lambda_i = 0$ *;*
t=0;
loop:
    P = solution of LSP by shortest path algorithm;
    if termination condition is satisfied halt;
    for each λ_i , $\lambda_i = \max\{0, \lambda_i + \theta_t \cdot (\sum_{e \in P} c_i^e - C'_i)\}$ *;*
    t++;
    update θ_t *s.t.* $\lim_{t \to \infty} \theta_t = 0$ *and* $\lim_{t \to \infty} \sum_{i=1}^{t} \theta_t = \infty$ *;*
end.

Figure 2.9: Algorithm: Sub-gradient method solving the MCSP.

Notice that it is still possible that the optimal solution to LMP is not a feasible

solution to MCSP according to the inequality. A strategy is to set a maximum number of iterations to save the run time. In each iteration step, the feasible solution to MCSP will be tracked.

2.3.2 The solution to MRRVC: the general case

Now we consider the general case where there may exist non-critical free vias, i.e., vias with more than one free neighbors. In the general case, we still hope to use the edge cost to estimate the number of vias in net *i* that is killed by the new net *m*, as theorem 1 does in the special case. A non-critical free vias will be killed if and only if all of its free neighbors are occupied. To count such a killed via, each of its free neighbor will contribute *1/Dof(v)* to the sum of the edge costs.

The algorithm of edge cost assignment in Figure 2.5 can be easily modified to consider these non-critical vias, as shown in Figure 2.10. The only modification is in the last line. Obviously, this new algorithm is equivalent to the algorithm in Figure 2.5 if the input routing graph meets the condition of the special case.

With this algorithm of edge cost assignment, the algorithm in Figure 2.9 can still be used to solve the MCSP in the general case. However, notice that in the general case, the sum of the edge costs along the new net *m* may overestimate the killed vias. For example, the following case will lead to the sum of the edge cost by 1: the new net passes two free neighbors that belong to two different vias with *DoF* 2. In this case, each in edge of the free neighbor will contribute 1/2 to the sum of the edge cost. However, the two vias are still alive as both of them have *DoF* 2. Although losing a free neighbor, they still have one free neighbor available for redundant via insertion.

```
Input: net i, the routing graph
Output: cost assignment

for each edge e in the routing graph
    cost(e) = 0;
for each live via v in i
     for each free neighbor n of v
          for each incident edge e of n
               cost(e)=cost(e)+1/Dof(v);
end.
```

Figure 2.10: The algorithm of edge cost assignment in the general case

To eliminate the overestimation, the term of $(\sum_{e\in P} c_i^e - C'_i)$ in the algorithm in Figure 2.9 can be replaced with *slack(i)*, where *slack(i)* is defined as $DV_i - C_i$, DV_i is the number of dead vias in net *i* after net m is routed, C_i is the maximum allowed dead vias for net *i*.

In fact, it can be proved that $(\sum_{e\in P} c_i^e - C'_i)$ = *slack(i)* in the special case: According to the notation definition in MCSP definition, $DV_i = KV_i + DV_i'$, $C_i' = C_i - DV_i'$. From theorem 1, $\sum_{e\in P} c_i^e = KV_i$. Therefore, $(\sum_{e\in P} c_i^e - C'_i) = KV_i - (CV_i - DV_i') = KV_i + DV_i' - CV_i = DV_i - C_i = slack(i)$. This equation indicates that the update of λ_i can actually be based on *slack(i)* whose calculation does not depend on the edge cost assignment, and the occurrence of *slack(i)* is a reasonable generalization. The algorithm in Figure 2.9 is modified to solve the MCSP problem in the general case, as shown in Figure 2.11. The only modification is at *slack(i)*.

cost assignment in Figure 2.10;
for each λ_i $\lambda_i = 0$ *;*
t=0;
loop:
P = solution of LSP by shortest path algorithm;
if termination condition is satisfied halt;
for ach λ_i $\lambda_i = \max\{0, \lambda_i + \theta_t \cdot slack(i)\}$ *;*
t++;
update θ_t *s.t.* $\lim_{t\to\infty} \theta_t = 0$ *and* $\lim_{t\to\infty} \sum_{i=1}^{t} \theta_t = \infty$ *;*
end.

Figure 2.11: Algorithm: Sub-gradient method solving the MCSP in the general case.

2.4 EXPERIMENTAL RESULTS

We implement the constrained maze routing algorithm in C++. The platform to run the experiments is a Xeon 3.4G dual-processor workstation with 2GB memory. Our router is a multi-layer detailed router. Each layer has a restricted direction: the odd layer is restricted to have x-axis edge only and the even layer only allows y-axis edge.

We perform the experiments on three circuits with different sizes. In circuit I, 96 nets are routed in a 40x40 grid; in circuit II, 348 nets are routed in a 120x120 grid; in circuit III, 650 nets are routed in a 200x200 grid. The size of routing grid is in the same order of magnitude as the one that typical industry detailed routers will apply to. The routing region is a 4-layer over-the-cell routing region. The odd layer is restricted to have x-axis edge only and the even layer only allows y-axis edge.

We set the constraint on the number of dead vias per net, and run the constrained maze router respectively: constraint D0 means no dead vias are allowed at all; constraint D1 means no more than one dead via per net is allowed; constraint D2 means no more than two dead vias per net are allowed. Then we run the conventional maze routing, denoted as C. Finally, we compare the number of feasible nets obtained by each run. Figure 2.12, Figure 2.13, and Figure 2.14 are the comparison of the experimental results. The experimental results show that our algorithm can always find routing solutions with higher rate of redundant via than conventional maze routing.

Compared with conventional maze routing, our algorithm also has reasonable run-time and good average wire length per net, as shown in Table 2.1. The run time increases up to 3.5x. However, the 3.5x time slowdown is mainly due to the strong constraint of no dead via allowed for any net, an extreme case that is supposed to be hard to find the feasible solution. The average wire length does not increase, furthermore.

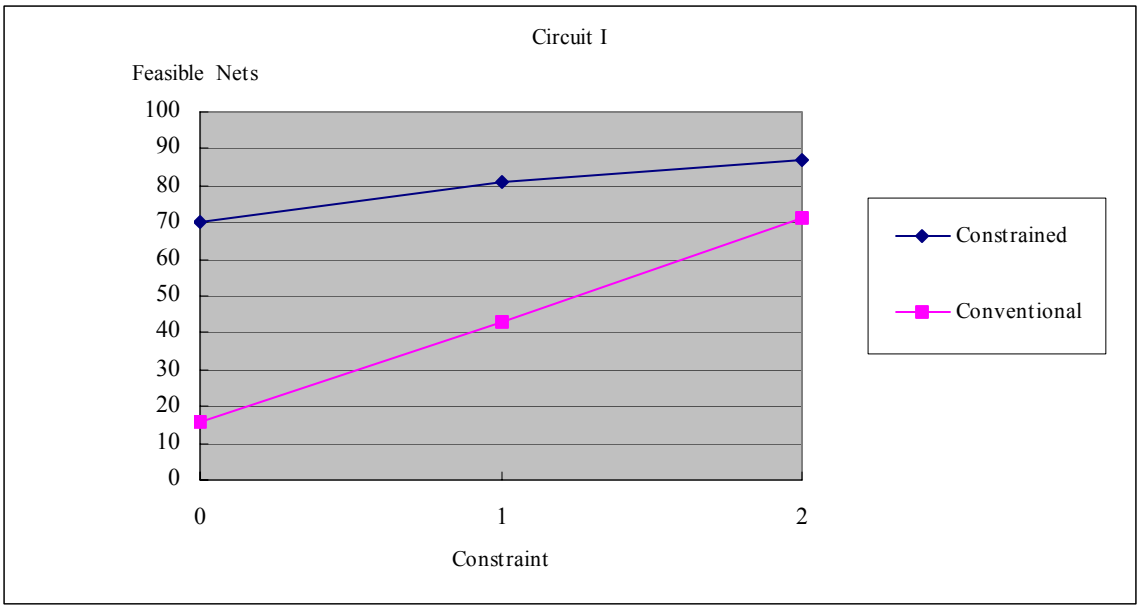


Figure 2.12: Comparison of the number of feasible nets in circuit I

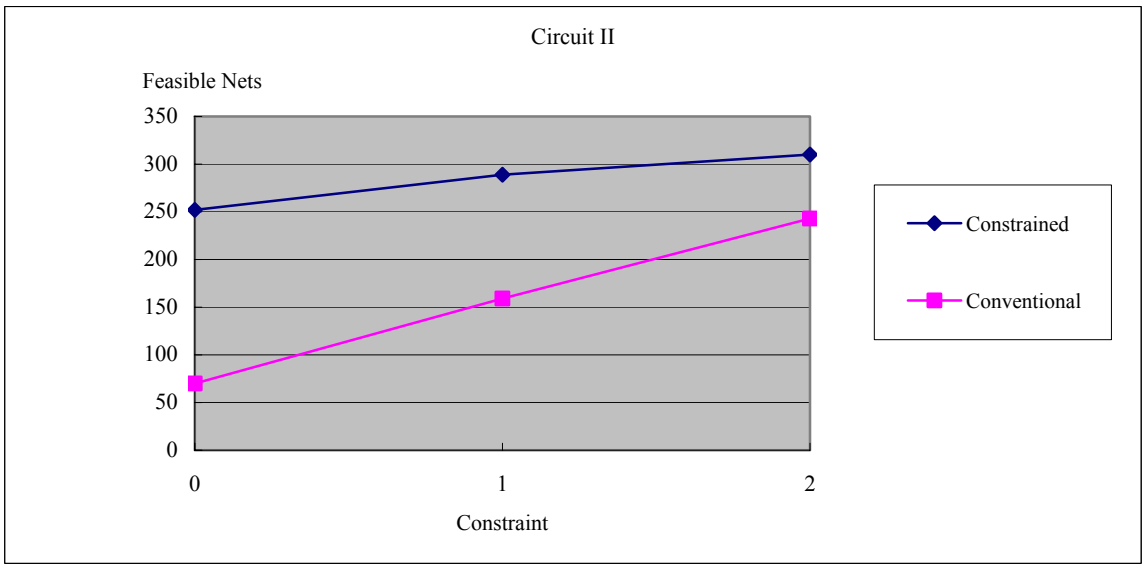


Figure 2.13: Comparison of the number of feasible nets in circuit II

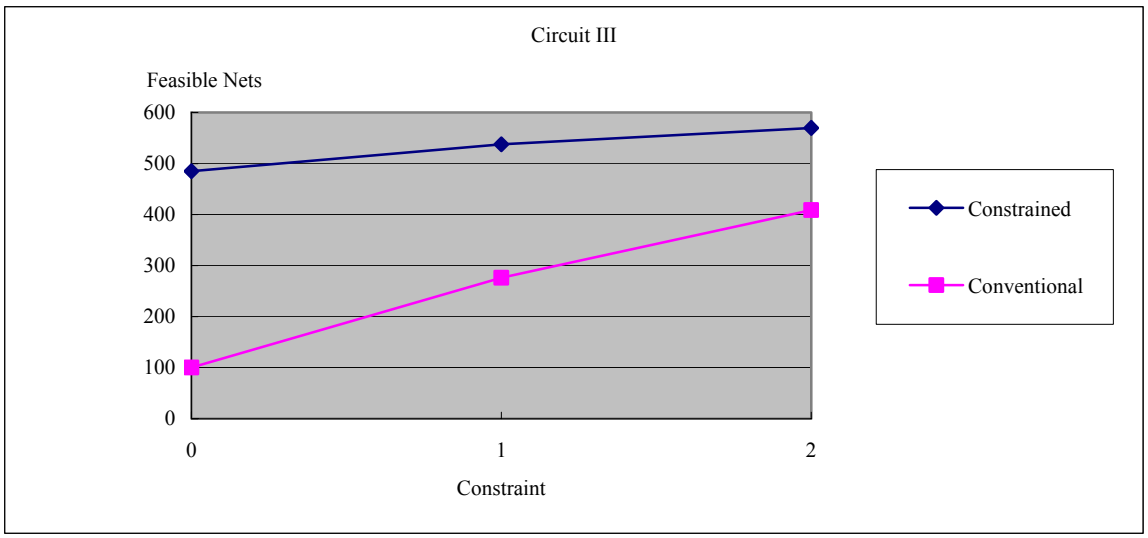


Figure 2.14: Comparison of the number of feasible nets in circuit III

Circuit	I				II				III			
Constraint	D0	D1	D2	C	D0	D1	D2	C	D0	D1	D2	C
Run time (sec)	1.4	1.02	0.83	0.5	55.2	44.66	35.81	16.11	353.82	260.24	187.47	101.75
Average WL	34.33	33.95	34.67	36.81	84.52	85.31	86.6	90.89	139.68	139.68	140.67	151.40

Table 2.1: Comparison of the run time and average wire length

2.5 CONCLUSION

In this chapter we present a constrained maze routing algorithm that can guarantee the redundant via insertion for each net, which is important to reduce yield loss caused by via failure in today's IC manufacturing. By assigning cost vector to each edge, the

problem of maze routing with redundant via constraints is first transformed to a multi-constrained shortest path problem, and then solved by Lagrangian relaxation technique. Experimental results show that our algorithm can find routing solutions with higher rate of redundant via than conventional maze routing.

Chapter 3: Shuttle Mask Floorplanning

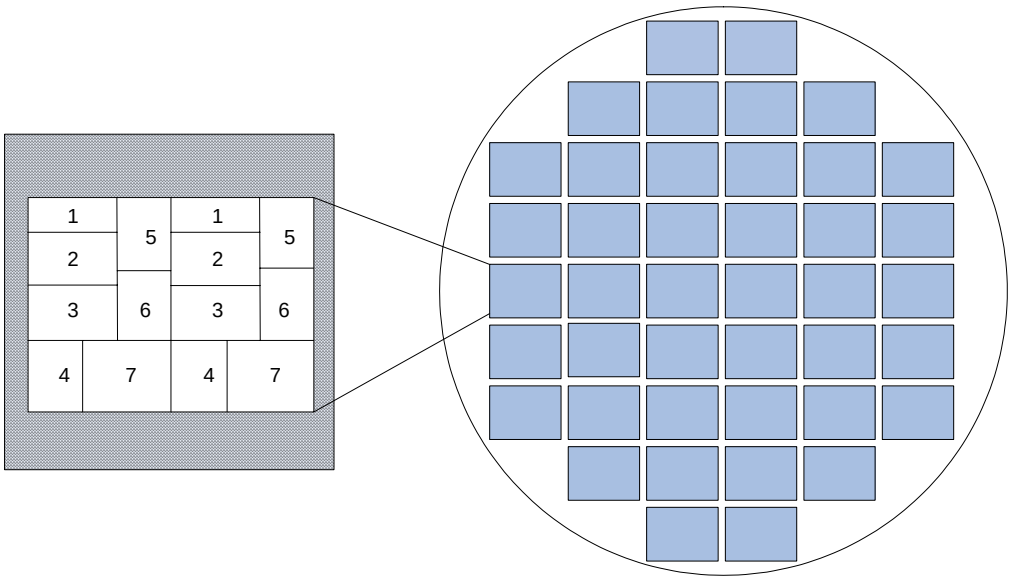


Figure 3.1. A shuttle mask and the projections on the wafer. Each small rectangle with a number represents a chip.

3.1 INTRODUCTION

Shuttle mask is an economical method to share the soaring mask cost for low volume designs by putting different chips on the same mask, as shown in Figure 3.1. In a simple cost model where each chip design is charged based on the area it occupies on the mask, the mask cost will be halved for each design if the mask is shared by two designs equally. It is noted that use of the shuttle mask may lead to extra overhead such as extra time to merge different data files and additional expense to cut out different chips from

one wafer. However, the total cost is still much lower than the cost of making multiple mask sets. Because of its cost advantage, shuttle mask service begins to proliferate. Chip designers can access the shuttle service provided by major foundries such as TSMC and IBM.

It naturally follows a floorplanning problem how to optimally pack different chips on the shuttle mask. Unlike traditional floorplanning problems in circuit design whose objective is to minimize the chip area and total wire length, shuttle mask floorplanning needs to handle objectives and constraints regarding cost and manufacturability in VLSI circuit manufacturing. These objectives and constraints may include: (1) area minimization to save mask cost; (2) die-to-die inspection constraint to improve the defect inspection; (3) wafer utilization to save wafer cost and chip production time, and (4) others, for example, die orientation constraint to guarantee the manufacturability. Therefore, shuttle mask floorplanning is distinguished from the traditional floorplanning problem, and has attracted interests of EDA community [19, 39, 40, 41, 42, 43, 44].

To our knowledge, Chen and Lynn published the earliest paper on shuttle mask floorplanning in early 2003 [39]. They only considered the area minimization objective that was actually a simplified version of classical floorplanning problem. The floorplan can be either slicing or non-slicing. Later Xu et al. [19] studied the minimum area floorplan problem with die-to-die inspection constraint that is important for defect inspection on mask. They used slicing floorplans. Around the same time appeared Andersson et al's work [42], in which they used a "grid" floorplan that tried to handle both the area minimization and wafer utilization maximization. Afterwards, Kahng et al. [43] also studied the problem of simultaneous area minimization and wafer utilization

maximization. They used non-slicing floorplans, and assumed the chip with varying width and height. Later, Kahng et al [44] considered another formulation of area minimization and wafer utilization maximization in which wafer utilization was represented as a constraint, instead of an objective. In this paper, they revisited "grid" floorplan and removed the questionable assumption of varying margin.

In this section, we present a simulated annealing based slicing floorplanner that can solve the problem of shuttle mask floorplanning with multiple optimization objectives and constraints simultaneously. We have the following contributions.

(1) Compared with the previous work, our work is the first complete work that can handle all objectives and constraints discussed above: area minimization, feature density optimization, wafer utilization maximization, die-to-die inspection constraint, and die orientation constraint. We will also show that how to extend our floorplanner to be "CMP-aware" in the next chapter that studies a class of post-CMP variation related problems.

(2) In wafer utilization maximization, our floorplanner assigns different dicing plans to different wafers, which can improve the wafer utilization compared with [44].

(3) Our floorplanner reduces area minimization with the die-to-die inspection constraint and the orientation constraint to an unconstrained area minimization problem. With the constraint removed, the new problem can be easily incorporated with other objectives.

This chapter is organized as follows. In Section 3.2, preliminaries of the floorplanner are introduced. Section 3.3 shows how to solve area minimization with the die-to-die inspection constraint and die orientation constraint. Section 3.4 studies

simultaneously area minimization and wafer utilization maximization. Section 3.5 shows experimental results.

3.2 PRELIMINARIES

Our multi-objective floorplanner for shuttle mask starts from the objective of area minimization, because it is a natural and important objective for shuttle mask floorplanning. Given a set of chips, a compact shuttle mask floorplan will have more projections on the wafer; it also allows more chips to be put on the shuttle as long as these chips can be packed in the frame of maximum printing field. The mask cost is reduced in both cases. Area minimization of shuttle mask floorplan is indeed a rectangle packing problem that is proved NP-hard [45]. However, by using the technique of simulated annealing (SA), there have been many floorplanners that can efficiently find a near-optimal solution [46, 47, 48, 49, 50].

Among these floorplanners, we choose Wong-Liu floorplanner [46] as the basis to start with. A major reason is that it uses the slicing structure as the topological representation of a floorplan. A slicing structure is obtained by recursively cutting a rectangle into smaller rectangles horizontally or vertically. To form a slicing floorplan, each chip will be put in an indivisible small rectangle called basic block. The topological structure of a slicing floorplan can be elegantly represented as a rooted binary tree, or equivalently, a normalized polish expression, as shown in Figure 3.2.

As the matter of fact, in the published work on shuttle mask floorplanning, grid floorplans, slicing floorplans, and non-slicing floorplans are all used. Grid floorplans are not preferred, however, as they often have large white space. Between slicing and non-

slicing structure, we prefer slicing structure because of its simple and nice tree structure and a smaller solution space [50]. Although a slicing floorplan is usually not as compact as a non-slicing counterpart for the same chip set, their results are close. In addition, Wong-Liu floorplanner provided a shape curve based floorplan realization with which we can get an elegant solution to handle the die-to-die inspection, as shown later.

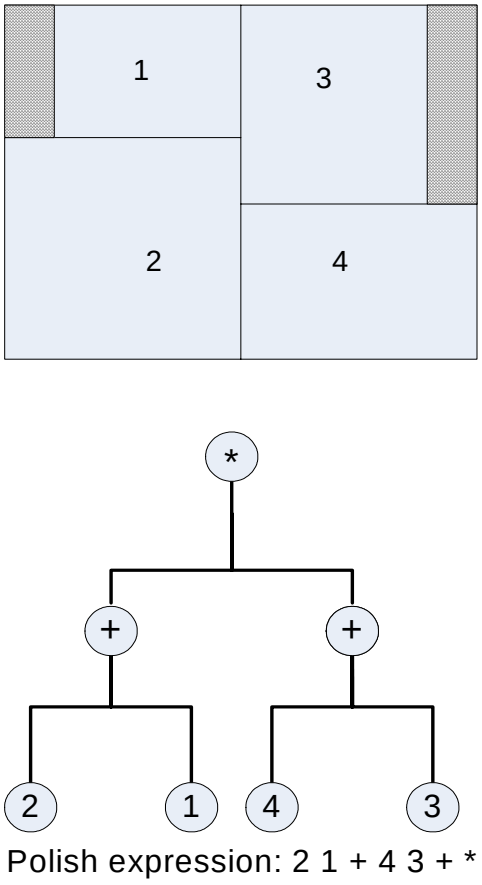


Figure 3.2: A slicing floorplan and its slicing tree representation, * represents vertical cut while + represents a horizontal cut. The shadow region refers to the white space.

3.3 Area Minimization with Die-to-die Inspection and Orientation Constraints

Defects may appear on the mask during the process of mask making. A defect is any flaw distorting the mask image from the original design, including extra chrome region such as chrome spots and chrome bridging between geometry, or extra clear areas such as pinholes and clear extensions [51]. In order to guarantee good manufacturability of VLSI circuits, defects on the mask must be carefully inspected and repaired before the mask is delivered. Die-to-die and die-to-database are two techniques for mask inspection. Die-to-die inspection compares two identical chip images at different positions on the mask. In contrast, die-to-database compares the chip image on the mask and the computer-generated image stored in the database. Die-to-die inspection has higher sensitivity to detect defects, as the defect is unlikely to appear twice at the same location of the chip images. However, chips under die-to-die inspection must appear pair-wise and be aligned horizontally or vertically on the mask for the sake of the requirement set by the inspection machine, which forms the die-to-die inspection constraint.

As the VLSI fabrication technologies continue to advance, chips with the strict transistor orientation is predicted to appear as well for the sake of great manufacturing benefits [30]. The transistor orientation will also impose the orientation of the chip, which refers to the orientation constraint.

Area minimization with die-to-die constraint on shuttle mask floorplan can be solved by using a merging method. In the merging method, a pair of identical chips to be aligned is first merged into a "super-block" that can have multiple shapes, as shown in Figure 3.3.

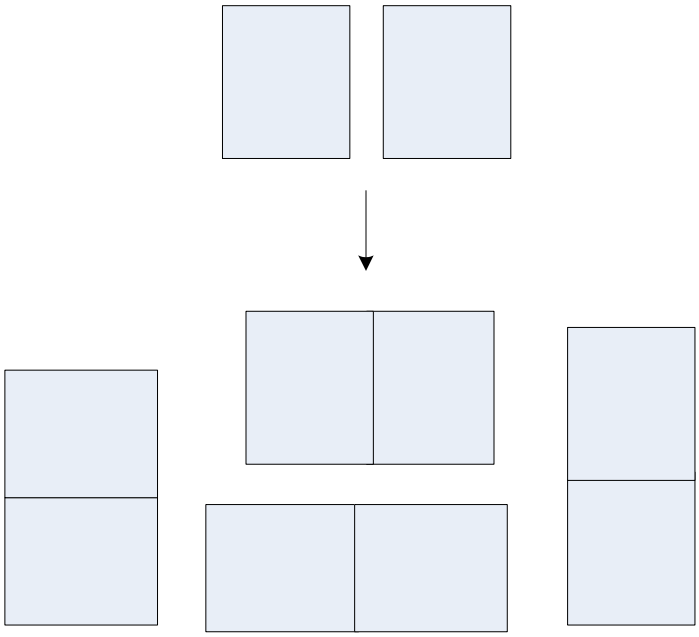


Figure 3.3: Multiple shapes of a super block

Multiple shapes of a block can be described by its shape curve, which is defined as the boundary of the feasible region of this chip. A point *(x, y)* is in the feasible region of this chip if and only if the rectangle with *(0,0)* as the left-lower corner and *(x, y)* as the right-upper corner can cover the chip. In a floorplan represented as a slicing tree, each leaf node will represent a basic block. The shape curve of the leaf node is determined by the allowed shapes of the basic block. The shape curve of an internal node is calculated by merging its two children nodes' shape curve. The shape curve of the root will determine the minimum area of the floorplan.

A nice property of shape curve is that, by initializing the shape curve of a basic block in different ways, we can easily guarantee that a shuttle mask is feasible with the die-to-die inspection and orientation constraints. For example, suppose we have a chip

with height h and width w. We can apply 4 possible combinations of die-to-die and orientation constraints on it: (1) the chip is under orientation constraint, but free of die-to-die inspection constraint; (2) the chip is free of both orientation and die-to-die inspection constraints; (3) the chip is under constraints of both orientation and die-to-die inspection; (4) the chip is free of orientation constraint but under die-to-die constraint. Here we refer a chip is under die-to-die inspection constraint to the scenario that it must be merged with another identical chip.

Figure 3.4 shows how the shape curve of the chip should look like in these situations. In any case, we can always reduce the die-to-die inspection or orientation constrained floorplanning problem into an unconstrained problem and solve it efficiently. In addition, this unconstrained problem is easy to incorporate with other objectives.

3.4 AREA MINIMIZATION AND WAFER UTILIZATION MAXIMIZATION

Most of the times, chips on wafer are cut out by a cutting saw that traverses the whole wafer horizontally or vertically. However, if shuttle mask is used, cutting out one chip in this way may destroy others, as chips on shuttle mask have different sizes and shapes. An example is shown in Figure 3.5.

Obviously, given the product volume of each chip and a shuttle mask floorplan, the less the chips are destroyed, the better the wafer is utilized. Consequently, we will have fewer wafers to be consumed, and save the wafer cost.

Therefore, wafer utilization, which can be defined as the reciprocal of the number of wafers required to meet the volume of each chip, becomes another important objective of shuttle mask floorplanning.

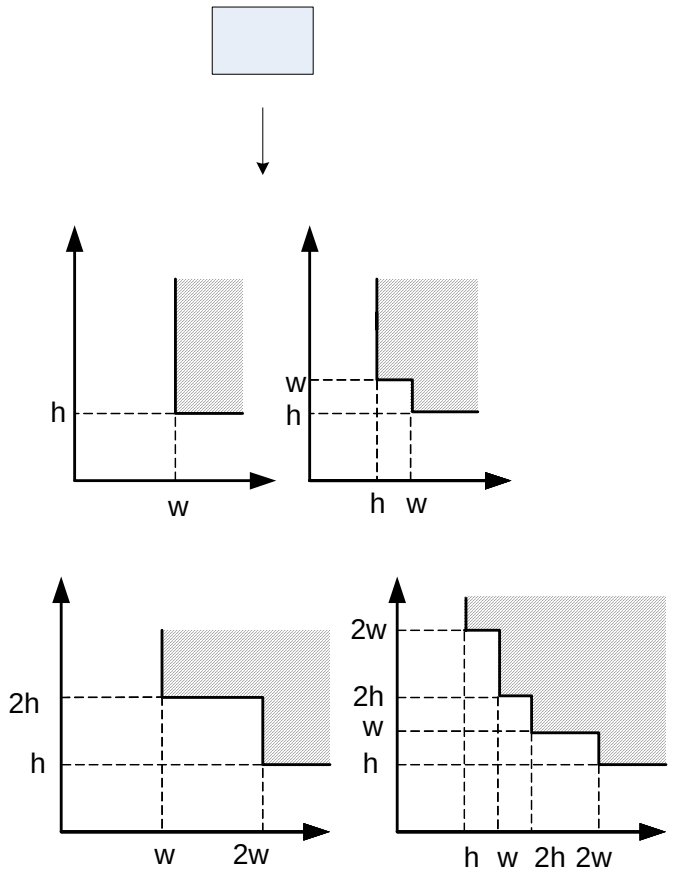


Figure 3.4: Different shape curves for a chip with height h and width w. From left to right, the shape curves are for a single chip with and without the orientation constraint, a pair of merged chips with and without orientation constraint respectively. The shadow region in each graph refers to the feasible region.

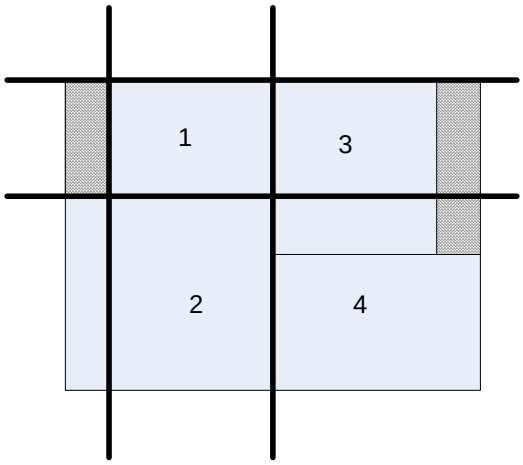


Figure 3.5: Wafer cutting. For simplicity only one projection of shuttle mask on the wafer is shown. Cutting out chip 1 will destroy chip 2 and 3.

As seen above, the wafer utilization is determined by the possibility of chip damage. The less the possibility, the higher the wafer utilization. The concept of "conflict" is useful to help understand the relation. Consider two chips A and B. Their positions on the shuttle mask will determine whether cutting out one will destroy another. The position of a chip can be represented by a pair of intervals: *(* [L_x, U_x], [L_y, U_y]), where L_x and L_y are the coordinates of the left-down corner; U_x and U_y are the coordinates of the right-up corner. Obviously, A and B can be cut out simultaneously without destroying each other if and only if (1) $[L_x^A, U_x^A] \cap [L_x^B, U_x^B] = [L_y^A, U_y^A] \cap [L_y^B, U_y^B] = \varnothing$, which means the chips' projections on x-axis and y-axis are not overlapped at all, or (2) $[L_x^A, U_x^A] = [L_x^B, U_x^B]$, which means A and B are aligned vertically, or (3) $[L_y^A, U_y^A] = [L_y^B, U_y^B]$, which means A and B are aligned horizontally. For any pair of chip not satisfy the above

conditions, we call them are in horizontal or vertical conflict. A shuttle mask floorplan with high wafer utilization is expected to have fewer conflicts among chips on the mask.

Efforts have been made in previous work to solve the wafer utilization maximization. [42] considered a "grid" floorplan for shuttle mask that reduced the possibility of chips' overlap on x-axis or y-axis, as shown in Figure 3.6. A shuttle mask will be partitioned into a grid first. Then each cell in the grid will be assigned to a chip. The grid structure prevents chips neither in the same row nor in the same column from conflicting. They studied the area minimization problem of such grid floorplan and its variants, and suggested a series of approximation algorithms. Their approach looks interesting theoretically. However, they didn't explicitly evaluate the wafer utilization of a grid floorplan, and no experimental results were reported to show the effectiveness of their approach. [43] was the first paper that explicitly evaluated the wafer utilization (yield in their paper). They used the non-slicing floorplan to represent shuttle mask (multi-project reticle in their paper). Given a shuttle mask floorplan, they defined H-conflict and V-conflict graphs to indicate the conflict relation between any two chips on the mask. An example is shown in Figure 3.7.

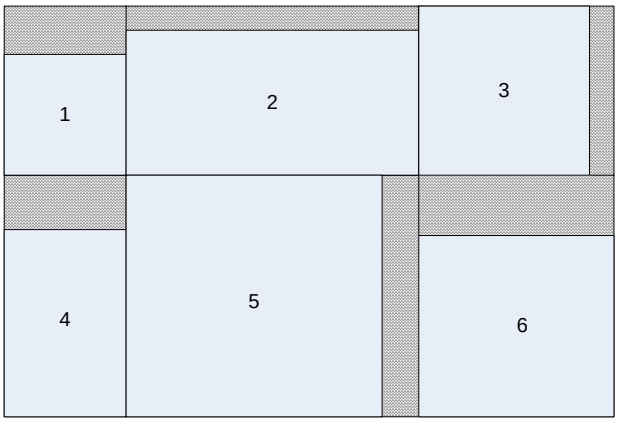


Figure 3.6: A grid floorplan

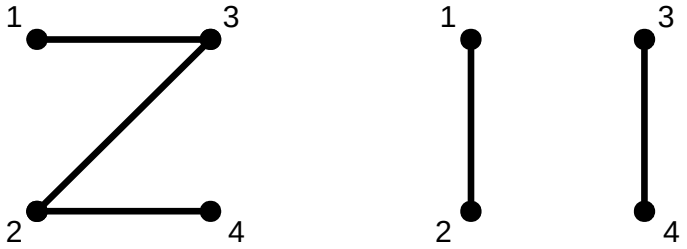


Figure 3.7: The H-conflict graph (left) and the V-conflict graph (right) for the floorplan in Figure 3.5.

A maximal independent set in H-conflict graph corresponds to a set of chips that can be horizontally cut at the same time. Assuming reticle projections were arranged as an RxT matrix, they assigned an independent set of H-conflict graph (ISH) to each row and an independent set of V-conflict graph (ISV) to each column. For the reticle projection at (i, j), the intersection of i-th row's ISH and j-th column's ISV would determine which chips to be cut out. Such an assignment of ISH and ISV was called a

"wafer dicing plan", as shown in Figure 3.8. The cost of a dicing plan was defined as the minimum number of wafers required to get the volume of all chips, which is reciprocal of wafer utilization in our paper. Given a shuttle mask floorplan, they proposed a non-linear programming formulation and several integer linear programming formulation to find an optimal dicing plan, and a simulated annealing heuristic to quickly find the near-optimal solution. Cost of a shuttle mask floorplan was the weighted combination of area and cost of its dicing plan.

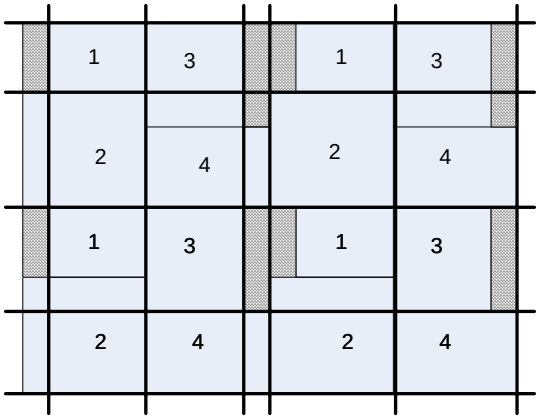


Figure 3.8. A wafer dicing plan. The reticle is shown in Figure 3.5. Assume its projections on wafer compose of a 2x2 matrix. Maximal independent sets {1,2} and {3,4} of H-conflict graph in Figure 3.7 are assigned to the first and the second row. Maximal independent sets {1,3} and {2,4} of V-conflict graph are assigned to the first and the second column.

However, a major problem appeared when they tried to reduce conflicts so as to save the cost of dicing plan. In their paper, they made an assumption that a chip could be cut out with variable margins from different reticle projections, as shown in Figure 3.9. Such variable margins will result in difficulties in packaging.

Later Kahng et al revisited the grid floorplan [44]. This time they removed the assumption of variable margins. In addition, wafer utilization appeared as a constraint, instead of an objective. Their problem was formulated as finding a grid floorplan with minimum area such that the wafer utilization was no less than certain value. They used branch-and-bound search to find the optimal solution. Experimental results showed improvements on both area and wafer utilization over [43].

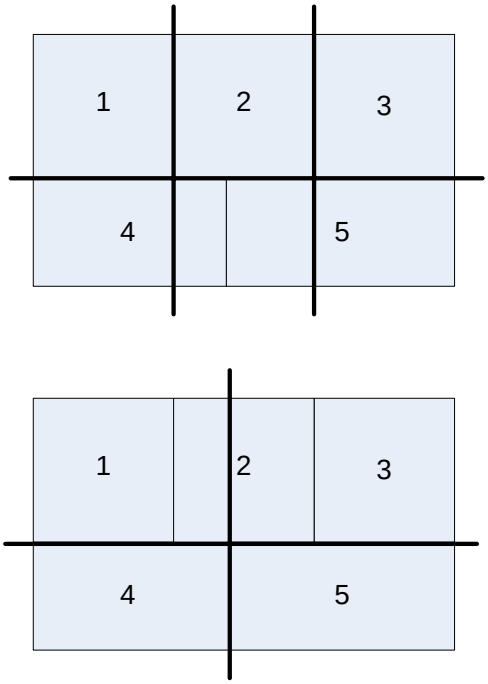


Figure 3.9. With variable margin assumption, chip 1 now can be cut out together with either {2, 3} or {4,5}. However, the two copies of chip 1 will have different size, for in the latter case chip 1 will have an extra margin.

Given a floorplan, our evaluation of the wafer utilization is different from the previous work: (1) unlike [43], we do not allow any chip to be cut out with any extra margin. For example, in Figure 3.9, the wafer dicing plan on the left is legal for us to cut

out chip 1, while the one on the right is illegal; (2) unlike [44], we still consider wafer utilization maximization as an objective, and calculate the weighted combination of area and wafer utilization, as the combination may reflect the total cost of mask and wafer. This combinational cost has a nice property that it can be easily adapted to different cost models of mask and wafer by adjusting weights of area and wafer utilization according to users' real situation; (3) unlike [43] that used the H-conflict and V-conflict graph of a floorplan separately, we use a single conflict graph which is sum of H-conflict and V-conflict graph to reflect the conflict relation among chips; (4) unlike [44] in which every wafer was cut in the same way, we may assign different dicing plans to different wafers.

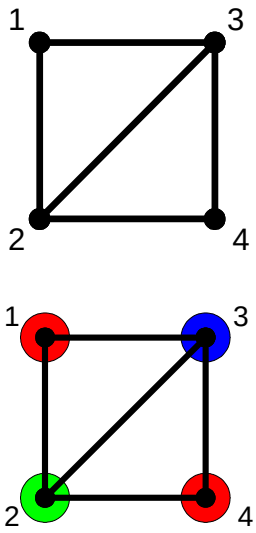


Figure 3.10. The conflict graph for the floorplan in Figure 3.5.

Consider the floorplan in Figure 3.5 whose H-conflict and V-conflict graph is shown in Figure 3.7, its conflict graph and a coloring scheme are shown in Figure 3.10. The optimal coloring scheme needs 3 colors, as there exist two 3-cliques in the graph but

no 4-clique exists. A coloring scheme will be: {1, 4} red, {2} yellow, and {3} blue. Obviously, chips with the same color are neither H-conflict nor V-conflict. We require that for any wafer, only chips with the same color can be cut out.

Three dicing plans for the floorplan in Figure 2 are shown in Figure 3.11. In the first dicing plan, red chip 1 and 4 are cut out simultaneously. However, yellow chip 2 and blue chip 3 have to be given up. These two chips will be cut out dedicatedly in the second and the third dicing plan.

A quick example will show how this strategy may improve wafer utilization. Assume we use the reticle in Figure 3.5 to print chips on wafer. The required volume of each chip is 240. The reticle is 4x with area 100mm x 132mm. The wafer has 200mm (8-inch) diameter. These data are all typical industry value. Assuming one corner of one reticle projection coincides with the wafer center and considering the round shape of the wafer, it is simple to calculate that there are at most 4x6=24 reticle projections. With the dicing plan in Figure 3.8, for each chip we can cut out 6 copies from the wafer. So we need 40 wafers to satisfy the volume requirement. However, with our dicing plans, from a wafer we can cut out 24 copies of each chip with the same color. The total number of required wafers thus becomes 240/24*3=30.

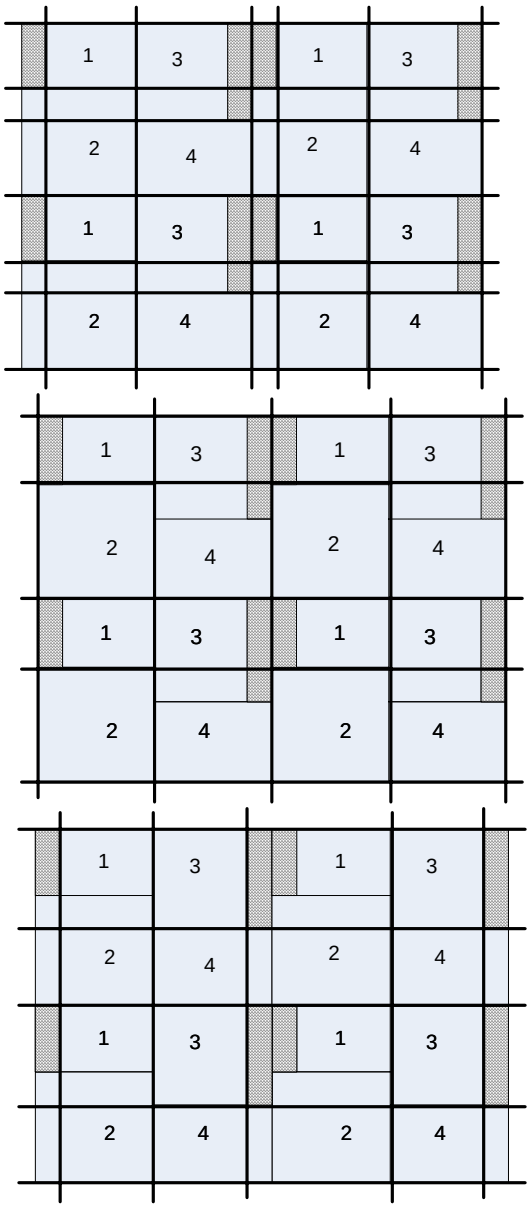


Figure 3.11. Dicing plans for the floorplan in Figure 3.5.

The calculation of wafer utilization is shown in Figure 3.12. Given a floorplan, we construct and color the conflict graph. Then we calculate the number of reticle

projections based on the sizes of the wafer and the floorplan. With the color number and the reticle projection number, we obtain the number of required wafers and wafer utilization.

```
input: a slicing floorplan F and the size of the
wafer
output: the number of required wafers

  construct the conflict graph G(F);
  color the conflict graph G(F);
  calculate the number of reticle projections on
wafer;
  calculate the number of required wafers;
```

Figure 3.12. The algorithm to calculate wafer utilization.

As graph coloring has been proved to be a NP-hard problem [52], we use a greedy coloring algorithms proposed by [53]. The algorithm is shown in Figure 3.13. The vertices in the graph are first sorted by the degree, i.e., the number of incident edge of each vertex. Vertices with higher degree will be chosen with higher priority to make the coloring in accordance to the coloring rule: if one vertex is painted with some type of color, any of its neighboring vertices cannot be assigned with this color any more. The procedure is repeated until all vertices are colored.

Our experiment shows that this greedy algorithm is a good approximation to the optimal coloring scheme of the conflict graph.

input: a general graph G
output: a coloring scheme

sort vertices of G by degree in the descending order;
for i = 1 to n do
assign the lowest indexed color c to v_i such that for any v_j adjacent to v_i , j<i, c is not assigned to v_j yet.

Figure 3.13. The greedy coloring algorithm

3.5 Experimental Results

We implement the shuttle mask floorplanner based on the Wong-Liu floorplanner [46]. The code runs on a Pentium-4 Linux workstation with a P4 2.4G Hz CPU and 1G DRAM.

The experiments on area minimization and wafer utilization maximization uses a data set derived from industry shuttle masks. This data set includes 12 chips with different sizes and shapes.

Table 3.1 compares the quality of the floorplans found by different weighted combinations of area and wafer utilization cost. (A, W) refers to the normalized weights for area (A) and wafer utilization (W). The number of wafers refers to the required number of wafers to cut out all chips. The wafer utilization is defined as its reciprocal. The smaller the number of required wafers, the larger the wafer utilization. The number of Projections refers to the maximum number of the reticle projections on wafer. Colors refer to the number of colors to color the conflict graph of the floorplan. White Space indicates how compact the shuttle mask floorplan is.

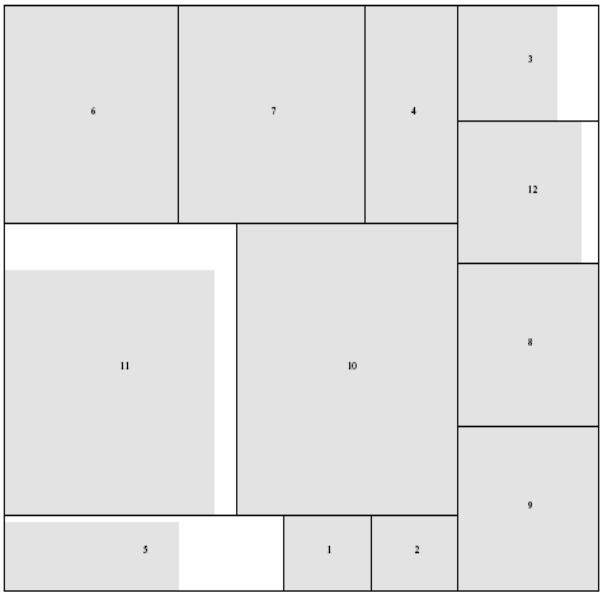
(A, W)	Number of Wafers	Number of Projections	Colors	White Space
(1, 0)	60	40	5	3.74%
(1, 0.1)	50	48	5	4.84%
(1, 0.5)	40	48	4	5.12%
(1, 1)	36	40	3	9.51%

Table 3.1: The comparison among different weighted combinations of area and wafer utilization

We can see the consistent trend that when the weight of wafer utilization increases, the wafer utilization is improved while the white space rate goes up. The floorplans for the best wafer utilization case and the best area case are shown in Figure 3.14 respectively. As the weights of area and wafer utilization are adjusted according to the different cost models of mask and wafer, the floorplan with the optimal total cost can be easily obtained from our floorplanner.

3.6 CONCLUSION

In this chapter, we investigate multiple objectives and constraints in shuttle mask floorplanning. We also present a simulated annealing based floorplanner to solve these objectives, constraints, and their combinations. Our floorplanner can be easily adapted to different cost models of mask and wafer manufacturing which may lead to different optimal solutions to different users. Experiments on industry data show very nice results.



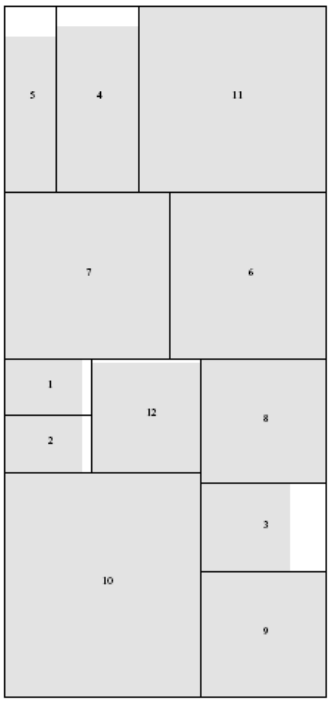


Figure 3.14: Floorplans for the best wafer utilization and best area.

Chapter 4: Studies on Optimization of Post-CMP Topography Variation

4.1 CMP TECHNOLOGY: A BRIEF REVIEW

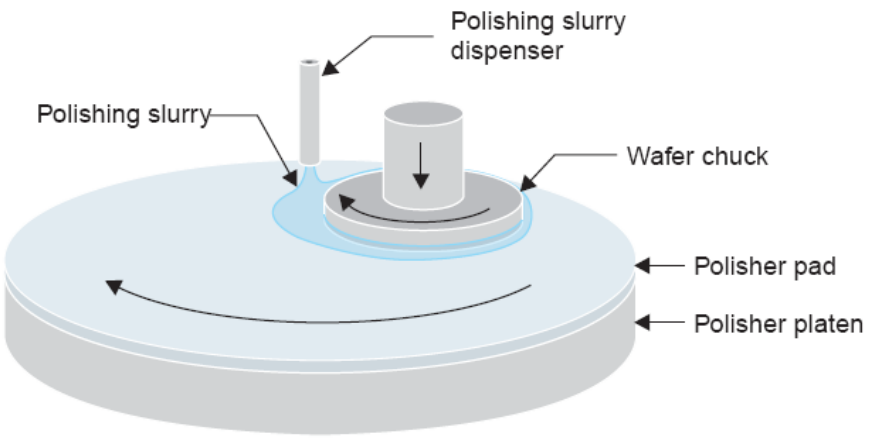


Figure 4.1: A CMP machine from [54].

Chemical mechanical polishing(CMP), also known as chemical mechanical planarization, is a manufacturing step to planarize wafer surface, as shown in Figure 4.1. In the CMP process, the wafer is polished rotationally through the polisher pad that is on top of a polisher platen. In other words, the polishing process is performed by the "mechanical" force. At the same time, the polishing slurry, an abrasive and corrosive "chemical" solution, is dropped on the polishing pad to accelerate the polishing process.

The CMP process is a necessary step for the technology nodes of nanometer scale, because the planarity of the wafer surface is important to control the depth of focus in the

next step's lithography, which will in turns affect the fidelity of the aerial image on the wafer.

At technology nodes of nanometer scale, there are three major types of CMP process applied on the wafer surface, depending on which layer is fabricated and which type of interconnect material is used. These CMP processes include oxide CMP, copper CMP, and shallow trench isolation(STI) CMP [55, 56, 57].

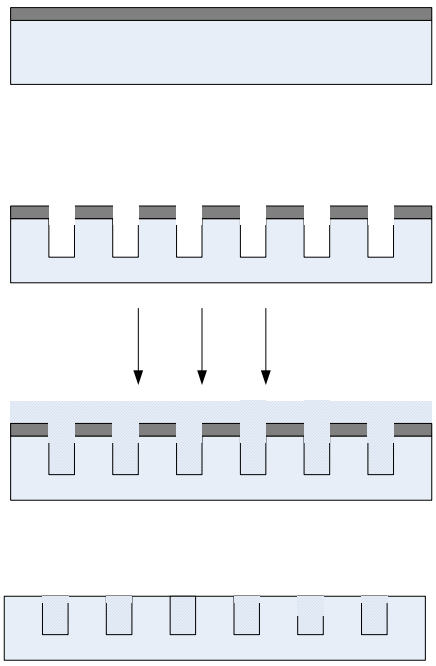


Figure 4.2: STI CMP. The dark features are nitride. The shadow features are oxide. The grey part is silicon substrate. For simplicity the last step of removing the left nitride layer is skipped.

Shallow trench isolation (STI) is the state-of-art isolation technique on the active layer at the 350nm technology node and below. In STI process, nitride is first deposited on the wafer surface to protect the active regions. Next, trenches are etched on those exposed region. Then oxide is deposited to fill these trenches. Since there is unwanted

oxide on top of the nitride-protected active regions, a CMP step is performed to remove the oxide and nitride layer until certain stop condition is reached. Finally, the rest nitride is stripped and the wafer surface now becomes a flat plane composing of all active regions isolated by the oxide, ready for the next step of gate patterning. These steps are shown in Figure 4.2.

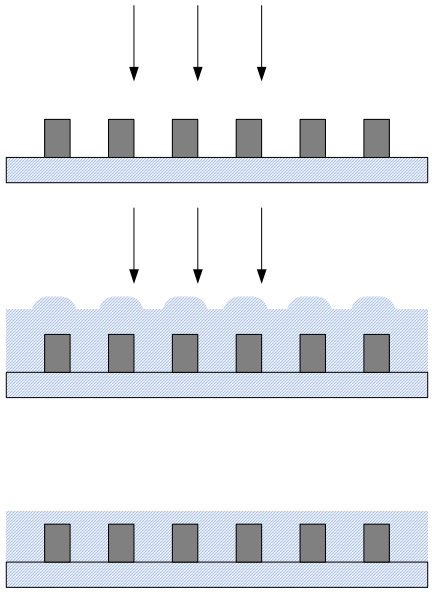


Figure 4.3: Oxide CMP for interlayer dielectric. The dark features are aluminum. The shadow features are oxide.

When aluminum is used for metal interconnect material, oxide CMP will be performed to polish the inter-layer dielectric. The manufacturing steps for an aluminum interconnect layer are as follows: (1) aluminum is deposited on the wafer surface to form the interconnect features; (2) oxide is deposited not only to isolate these interconnect features but also to form the inter-layer dielectric; (3) excess oxide is removed by CMP. Afterwards, the flat wafer surface is ready for the next layer to be manufactured, as shown in Figure 4.3.

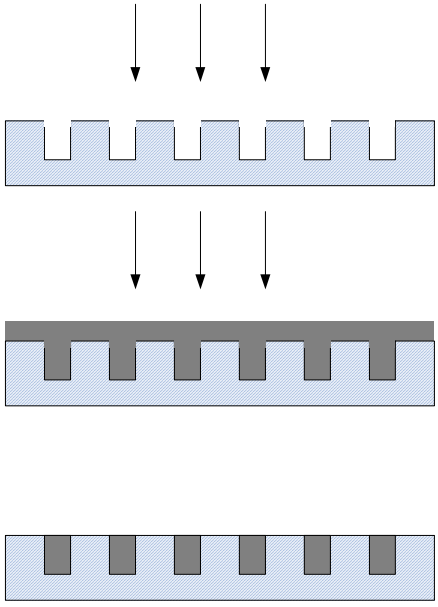


Figure 4.4: Copper CMP for interlayer dielectric. The dark features are copper. The shadow features are oxide.

Copper becomes the dominant metal material for interconnects since 130nm technology node and below because of its higher speed and less power consumption. However, because of the different physical properties of copper from aluminum, when copper is used, the interconnect layer is manufactured in another way called damascene process[26]. The main steps in this process are as follows: (1) interconnect region are etched on the oxide layer; (2) copper is deposited to fill these regions to form the interconnect features; (3) copper CMP is performed to remove the unwanted copper, as shown in Figure 4.4.

For simplicity we show a perfectly flat surface after CMP in Figure 4.2, 4.3, and 4.4. This is not true, however, in reality. After the CMP process, the surface will have topography variation. A lot of studies have been conducted to model and optimize the

post-CMP topography variation. These studies show that the variation is highly correlated with the feature density distribution of the layout [58, 59, 60]. Here the density is defined as follows: given a region, the density refers to the area covered by the feature divided by the total area of this region. In addition, the variation can be optimized by changing the feature density [61, 62, 63, 64, 65, 66, 67, 68, 69]. In the next section we will discuss how such a model can be used in the case of oxide CMP and STI CMP, and show how to realize optimization based on this model.

4.2 POST-CMP TOPOGRAPHY VARIATION: MODELING AND OPTIMIZATION

In this section, we focus on a semi-empirical model proposed by Stine et al [60] to estimate the post-CMP topography variation in oxide CMP. This model is well accepted and widely used, because it is inexpensive to compute, easy to calibrate, and yield reasonably accurate results. In this model, oxide thickness z at location (x,y) satisfies the following conditions:

$$z = \begin{cases} z_0 - [K_i t / \rho_0(x,y)] & t < \rho_0 z_1 / K_i \\ z_0 - z_1 - K_i t + \rho_0(x,y) z_1 & t > \rho_0 z_1 / K_i \end{cases} \qquad (1)$$

where

K_i: blanket oxide polishing rate;

z_0: thickness of oxide deposition;

z_1: initial step height;

t: total polish time;

$\rho_0(x,y)$: initial oxide pattern density before CMP.

The model indicates that after the polishing process lasts long enough, the thickness of the oxide is linearly proportional to the initial oxide pattern density. By

discretizing the layout into grids of small squares called cells, the initial oxide pattern density is represented by a matrix $\rho_0(i,j)$. Considering the deformation of the polishing pad during polishing, Ouma et al [58] derived the initial oxide pattern density, also known as effective density, from the feature density of the underlying layout using the following equation:

$$\rho_0(i,j) = IDFT[DFT[d(i,j) \cdot DFT[f(i,j)]] \qquad (2)$$

where *DFT* and *IDFT* are discrete Fourier transformation and its inverse operation respectively[70], and $d(i,j)$ is the feature density distribution represented by a matrix. The oxide CMP process is thus modeled as a 2-D low pass filter of feature density distribution $d(i, j)$ by the function f.

Tian et al [65] gave the following approximation of $f(x,y)$:

$$f(x,y) \approx c_0 \exp[c_1(x^2 + y^2)^{c_2}] \qquad (3)$$

where constants c_0, c_1 and c_2 are calibrated for any specific process.

Because the topography variation is proportional to the oxide pattern density and the oxide pattern density is determined by the feature density, we can reduce the variation by inserting dummy features into the layout to change the feature density. We should emphasize this is a very important observation and it is the basis of developing optimization work on post-CMP topography variation.

Tian et al [65] rewrote equation (2) as a convolution:

$$\rho_0(i,j) = \sum_{i'=i-L}^{i+L} \sum_{j'=j-L}^{j+L} [(x_{i'j'} + x_{i'j'}^{0}) \cdot f(i'-i, j'-j)] \qquad (4)$$

where $x_{i',j'}$ is the variable representing the amount of dummy feature to be inserted at postion *(i', j')*, and $x^0_{i',j'}$ is the feature density of cell (*i, j*). They also presented a simple and elegant LP formulation to describe the problem of topography variation minimization as follows:

$$\text{Minimize} \qquad \rho^H - \rho^L \qquad (5)$$

$$\text{subject to} \qquad \begin{aligned} &0 \le \rho^L \le \rho_0(i,j) \le \rho^H \le 1 \\ &0 \le x_{i,j} \le x_{i,j}{}^a \end{aligned}$$

where ρ^H and ρ^L are auxiliary variables and x^a_{ij} is the maximum capacity for dummy features at cell (*i,j*).

In practice, the total amount of dummy feature inserted is also an important concern, because a smaller amount usually leads to higher polish rate and less impact on users' design. Tian et al[65] also gave the following ranged-variation formulation that can be applied to the case where a smaller amount of dummy feature is preferred and a near optimal variation is acceptable. The formulation is as follows:

$$\text{Minimize} \qquad \sum_{i,j} x_{i,j} \qquad (6)$$

$$\text{subject to} \qquad \begin{aligned} &0 \le \rho^L \le \rho_0(i,j) \le \rho^H \le 1 \\ &\rho^H - \rho^L \le \varepsilon \\ &0 \le x_{i,j} \le x_{i,j}{}^a \end{aligned}$$

where ε is the variation budget parameter that describes how much variation can be afforded in order to get the minimum dummy fill. Obviously, the budget must be larger than the solution to (5).

Although the above model is for oxide CMP, a recent study by Beckage et al [71] showed that it can also be used for topography variation after STI CMP. They provide an excellent solution that treats the two stages in STI CMP separately with background and regional dummy fills by taking advantage of the oxide fill characteristics before CMP. With the background dummy fill providing mostly nitride density only, the dummy fill problem for STI becomes an oxide CMP problem again, which can be solved optimally with LP as described previously.

In the rest of this chapter, we will show three optimization study of post-CMP topography variation based on the low pass model and its related work discussed as previously. Section 4.3 shows how to enhance the shuttle mask floorplanner in Chapter 3 to be CMP aware. Section 4.4 focuses on a fast incremental algorithm to speed up the calculation of a set of layout that utilizes the similarity of these layouts. Section 4.5 demonstrates a novel quadratic formulation of post-CMP topography variation based on (5) for minimization of image distortion by defocus.

4.3 CMP AWARE SHUTTLE MASK FLOORPLANNING

As we discussed in chapter 3, nowadays the shuttle mask has become an economical method to share the soaring cost among different chips. In that chapter, we demonstrated a shuttle mask floorplanner to handle multiple objectives and constraints such as area, wafer utilization, and die-to-die inspection constraint. In this chapter, we will enhance the floorplanner to optimize a new objective: minimization of post-CMP topography variation. In other words, this enhanced floorplanner becomes "CMP aware" now. To our best knowledge, we present the first study on this topic.

It is an important feature to enhance the shuttle mask floorplanner to be CMP aware, because when the positions of the chips on the shuttle mask change, so does the feature density distribution of the whole mask. Hence, the change of the feature density distribution will result in the change of the post-CMP topography variation. Different from conventional "intra-chip" optimization on post-CMP topography variation that is performed by adding dummy features in the circuit layout, our optimization targets at the best "inter-chip" solution.

Because integrated circuits are fabricated layer by layer, a single shuttle mask floorplan must be used for the whole mask set. However, most of the time this floorplan is not optimal for all layers with respect to post-CMP topography variation. In our CMP aware shuttle mask floorplanner, we focus on the active layer and STI CMP. This is because features on the gate layer are the finest ones in circuits. Fabrication of gate layer is the most challenging step, and this step is done right after STI.

4.3.1 The three-step procedure

The objective of CMP aware floorplanning is a weighted combination of area and post-CMP topography variation with respect to STI CMP. We propose a 3-step procedure to solve the problem as follows.

First, based on the low-pass filter model described in Section 4.2, we propose three predictive functions to foresee the variation and guide the floorplanner. Specifically, we run the simulated annealing to search for the optimal objective: to minimize the weighted sum of area and post-CMP topography variation. At each simulated annealing (SA) search move, the slicing tree is realized to its minimum area

floorplan. For this floorplan, the predictive function is evaluated. Notice that we cannot call LP solver in the SA search because of the expensive computational cost of the LP method. The predictive function must be fast.

Because our shuttle mask floorplanner employs slicing tree as the topological representation, the chip may be movable in its enclosing rectangle. Therefore, in the second step of our procedure, the best result obtained in the first step is then further improved by sliding each chip in the boundary.

In the final step, given that the optimal floorplan and the optimal position of each chip are determined, we call the LP solver to get the optimal amount of dummy features to be inserted. Since LP method is called only once at this stage, its computation expense is acceptable. A pseudo code describing the algorithm is in Figure 4.5.

```
x =initial floorplan;
SA search with cost function f(x) =
area(x) + w p(x) ;
// p(x) is the predictive function; w is the
weight
sliding(x) to improve f(x);
for the best solution bestx, doing dummy
insertion;
```

Figure 4.5: The 3-step procedure to find the optimal solution

4.3.2 Predictive function

The predictive function in our simulated annealing is a weighted sum of area and a predictive function. We develop three functions to predict the topography variation in

the SA search: *MaxDiff*, *SDH*, and *NSDH*. For these three functions, the less the value, the better the variation. These notations are used in the following discussion:

$D^0 = (d^0_{i,j})$: the feature density matrix without dummy insertion.

$\mathrm{P}^0 = (\rho^0_{i,j})$: the effective density matrix without dummy insertion, which is derived from D^0 according to equation (2).

$C = (c_{i,j})$: the capacity matrix.

The function *MaxDiff* is defined as:

$$MaxDiff = \max\{\rho^0_{i,j}\} - \min\{\rho^0_{i,j}\} \qquad (7)$$

This function represents the maximal difference between the effective densities of cells in the floorplan. By using the *MaxDiff* function, we actually use the topography variation before the dummy feature insertion to predict the topography variation after the dummy feature insertion. This function is necessary when the capacity matrix C is a sparse matrix, which corresponds to the case that chips on the mask have strong restriction on dummy insertion. For example, sensitive circuits hand crafted by designers, like analog circuits, forbid automatic dummy insertion in the mask floorplanning stage after circuit tape-out.

The prediction of *MaxDiff* is not always reliable because the function ignores the dummy feature insertion. An alternative function *SDH*, representing "sigma delta height", is proposed to improve the accuracy of the prediction. It is defined as:

$$SDH = \sum (1 - c_{i,j})(\rho^0_{i,j} - \min\{\rho^0_{i,j}\}) \qquad (8)$$

The definition of SDH is based on the following considerations: (1) we expect a cell with large variation to have large capacity, which implies more flexibility in

adjusting its feature density; (2) we expect the total weighted variation to be small, which suggests that the current floorplan is flat.

We also consider the case where variation budget is imposed and minimum amount of dummy fill is desired. According to equation (4), the effective density at cell *(i,j)* is most impacted by the feature density at cell *(i,j)*. Therefore, to achieve the objective of minimum dummy fill, a natural idea is to add dummy features directly to the cells with low effective density as much as possible. High capacity is thus preferred at the cells. In addition, large white space is not preferred, because cells in the white space also need filling. More such cells may indicate more dummy features to be inserted.

Therefore, we further modify *SDH* to get the third function *NSDH*, which stands for "new sigma delta height". This function is defined as:

$$NSDH = \sum (2 - c_{i,j})[1 + (\rho^0_{i,j} - \min\{\rho^0_{i,j}\}) / (\max\{\rho^0_{i,j}\} - \min\{\rho^0_{i,j}\})] \qquad (9)$$

The motivation is that the previous function *SDH* is not ideal to locate the floorplan with smaller white space. This is because the capacity of white space cell is 1, and thus does not contribute to the function value. In addition, *SDH* is not ideal for cells with minimum effective density for the same reason. We normalize the variation of each cell related to the minimum effective density in order to make a fair comparison between different floorplans. Without normalization the function may lead the search to the objective of minimum variation, a deviation from the objective of minimum dummy fill that we actually desire.

4.3.3 Experimental Results

When implementing the CMP aware feature in our shuttle mask floorplanner, we use FFTW3.0.1 [72] to compute Fourier transformation. In the final stage of the 3-step process, we use CPLEX as the LP solver[73].

Table 4.1 shows the comparison among different cost functions. Colum WS represents the white space rate. Column VwoD represents minimum variation without dummy insertion. The unit of the variation is angstrom. ColumnVwithD represents the minimum variation with dummy insertion, which is the topography variation from solving LP with the objective of minimum variation, as illustrated in equation (5). DAmount represents the minimum dummy fill amount obtained by solving the LP with the objective of minimum fill, as illustrated in equation (6). The value unit does not matter. So we skip it. The variation budget in the LP is obtained by rounding the minimum topography variation in the previous column to the next 10's. For example, in the case of area+ SDH, 64 are rounded to 70 to form the minimum dummy fill problem.

The results show that the predictive functions serve pretty well in variation optimization and minimum dummy fill. The variation is improved by around 30% in all of the three functions. With the same amount of dummy feature insertion, area+NSDH obtains a little larger variation than the results of the area+SDH. However, this function obtains the minimum white space among the three results as we expect. If we consider all three metrics of area, topography variation, and amount of dummy feature insertion, area+NSDH performs the best. Figure 4.6 shows the floorplan obtained by area+NSDH.

Function	WS	VwoD	VwithD	D Amount
area only	2.82%	818	92	340
area+MaxDiff	6.87%	612	67	338
area+SDH	8.27%	588	64	298
area+NSDH	6.04%	751	67	298

Table 4.1: Comparison among different cost functions

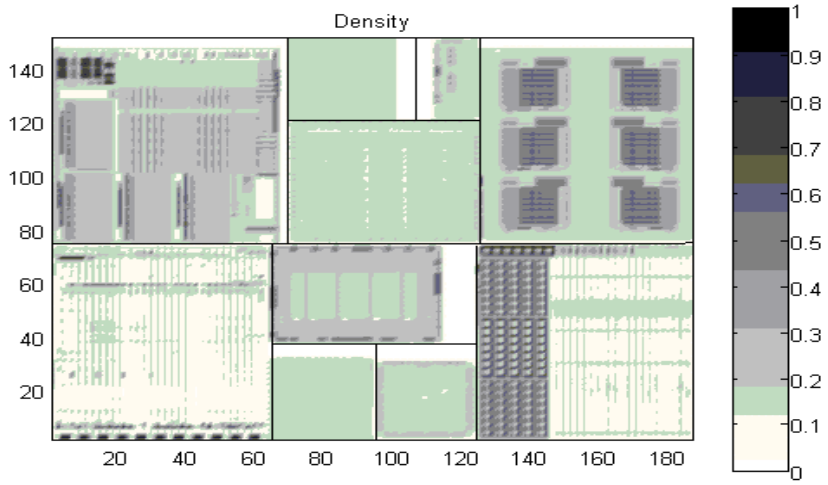


Figure 4.6: A shuttle mask floorplan by area+NSDH

4.4 A FAST AND EXACT INCREMENTAL ALGORITHM FOR COMPUTATION OF POST-CMP TOPOGRAPHY VARIATION

In the previous section, we have studied the problem of CMP-aware shuttle mask floorplanning. In the second step of the algorithm, given a slicing shuttle mask floorplan, we tried to move each block within its enclosing rectangle in order to get the optimal position with respect to post-CMP topography variation. In this section, we present a fast incremental algorithm that can quickly determine such an optimal position. The problem

is formulated as a single-block positioning problem (SBPP). By applying the linear and the shift properties of the convolution to the incremental layout, our algorithm only requires a simple *O(n)* matrix addition, rather than the *O(nlogn)* FFT operation in loop iteration, and thus saves much time. The experimental results show a consistent 6x to 9x times speedup compared to the non-incremental counterpart. Another advantage of this algorithm is that, it is easy to generalize this algorithm into multi-block positioning problem (MBPP) and apply to CMP aware shuttle mask floorplanning.

0.68	0.52	0.35	0	0
0.13	0.03	0.24	0	0
0.26	0.62	0.55	0.13	0.2
0.64	0.33	0.35	0.78	0.33
0.11	0.06	0.15	0.36	0.47

0.68	0.52	0.35	0	0
0.13	0.03	0.24	0.13	0.2
0.26	0.62	0.55	0.78	0.33
0.64	0.33	0.35	0.36	0.47
0.11	0.06	0.15	0	0

Figure 4.7: Topography variation will change as a block is moved within its range. The topography variation on the left side is 5.8 and the one on the right is 7.0 after normalization.

Our algorithm is still based on the low pass model introduced in Section 4.2, which describes the mathematical relation between the oxide thickness after CMP

process and the feature density distribution. We notice that the post-CMP topography variation will change as one block of the whole layout has the flexibility to move around, shown in Figure 4.7. A single-block positioning problem (SBPP) arises in this situation: what is the optimal position for this movable block to minimize the post-CMP topography variation?

This problem is important because its solution can be used to solve other complicated problems, for example, shuttle mask floorplanning. The solution to shuttle mask floorplanning is often a partition-based floorplan. Here partition-based floorplan refers to a floorplan that is partitioned into n rectangles each of which is assigned to a chip. For example, both the grid floorplan in [44] and the slicing floorplan in [40, 41] are partition-based, as shown in Figure 4.8. For these partition-based shuttle mask floorplans, multiple chips are free to move within its own enclosing rectangle, because there is no connection among these chips. The solution to our single-block positioning problem can be generalized to decide the optimal positions of the multiple movable chips in order to achieve the minimum post-CMP topography variation.

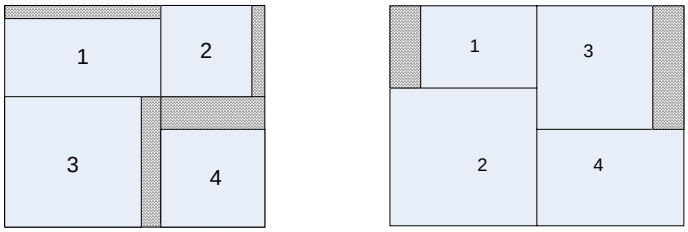


Figure 4.8: A grid shuttle mask floorplan and a slicing shuttle mask floorplan respectively.

4.4.1 The Single-block Positioning Problem (SBPP)

The single-block positioning problem (SBPP) is formulated as follows:

*Given a layout L represented as an M*x*N density matrix D, a block B represented as a p*x*q sub-matrix, and a range R represented as a pair of the intervals ([x, x+p+s-2], [y, y+q+t-2]. The block B can be freely moved within the range. The low-pass filter function f(x,y) is also given. Determine the optimal position of B such that the topography variation is minimized.*

The parameter (*M, N, p, q, x, y, s, t*) and the low-pass filter function determine an instance of SBPP. (*M, N*) and (*p, q*) represent the size of the matrix and the size of the sub-matrix respectively. (*x, y*) defines the starting point of the range; *s* and *t* represent the vertical and horizontal positions where block *B* can be possibly put. Figure 4.7 provides such an example. *D* is the 5x5 matrix, i.e. *M=N=5; B* is the 3x2 matrix, i.e., *p=3, q=2.* In this figure, we have *B*[*0,0*]*=0.13, B*[*0,1*]*=0.2*, etc. In this figure, *B* can be freely moved in the enclosing rectangle vertically with three possible positions. While *B* is fixed horizontally. Therefore, *s=3* and *t=1*. The enclosing rectangle starts at *D(0, 3)*. Therefore, the range is defined as *([0, 4], [3, 4]).*

4.4.2 A Simple Algorithm Solving SBPP Problem

A simple algorithm that directly uses the low pass model to solve the SBPP problem is shown in Figure 4.9. This algorithm calculates the topography variation of each position, and keeps track of the best one until the loop is finished. Since the dimension of the layout keeps unchanged during the loop, $DFT(f(i,j))$ can be pre-computed to save the time. Therefore, in each iteration step we need one DFT and one IDFT operation. If we use the fast Fourier transformation, the complexity of the algorithm is $O(s \cdot t \cdot n \log n)$.

4.4.3 The Incremental Computation of Topography Variation

We notice that the layouts before and after block *B* is moved are almost the same. This fact inspires us to find out an efficient algorithm to incrementally compute the topography variation. First, we rewrite (3) into the form of convolution:

$$\rho(i, j) = D(i, j) \otimes f(i, j) \qquad (5)$$

Then we decompose *D* into two matrices *X* and *D-X*, as shown in Figure 4.10. Obviously, when block *B* is moved, *D-X* keeps constant while *X* is changing. We use *C* to represent this constant matrix.

```
//input: a MxN matrix D, a pxq sub-matrix B, a range
([x, x+p+s-2], [y, y+q+t-2], the low pass filter
function f(i,j).
//output: the optimal position of B stored in
BestPostion

F = DFT(f); //discrete Fourier Transformation
BestTP = infinity;
BestPosition = (x,y);
for (i=0; i<s; i++)
      for (j=0; j<t; j++)
            update D according B's current position at
(x+i,y+j)
            rho = IDFT(DFT(D).F) // . represents dot
product
            TP = max {rho} - min {rho}
            if TP < BestTP
                  {
                   BestTP = TP;
                   BestPosition = (x+i,y+j);
                  }
```

Figure 4.9: the SBPP algorithm that directly uses the low pass model.

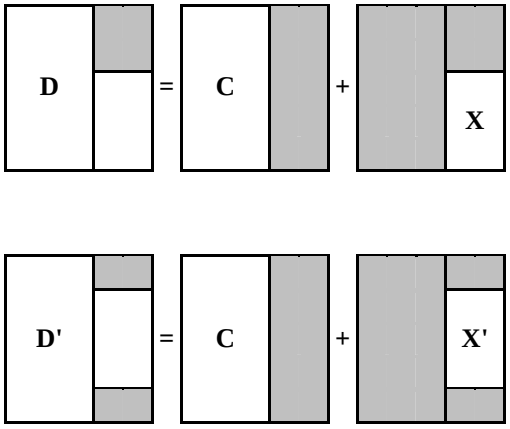


Figure 4.10: The density matrix *D* in Figure 2 can be decomposed into sum of two matrices *C* and *X,* where *C* = *D*-*X* is constant and *X* changes to *X'* as block *B* moves up.

According to the linear property of convolution [70], the topography matrices before and after *B* is moved are as follows respectively:

$$\begin{aligned}
&\rho(i,j)\\
&= D(i,j) \otimes f(i,j)\\
&= (D(i,j) - X(i,j)) \otimes f(i,j) + X(i,j) \otimes f(i,j)\\
&= C(i,j) \otimes f(i,j) + X(i,j) \otimes f(i,j)\\
&= K(i,j) + X(i,j) \otimes f(i,j)
\end{aligned} \tag{6}$$

$$\begin{aligned}
&\rho'(i,j)\\
&= D'(i,j) \otimes f(i,j)\\
&= (D'(i,j) - X'(i,j)) \otimes f(i,j) + X'(i,j) \otimes f(i,j)\\
&= C(i,j) \otimes f(i,j) + X'(i,j) \otimes f(i,j)\\
&= K(i,j) + X'(i,j) \otimes f(i,j)
\end{aligned} \tag{7}$$

Therefore, we only need to compute the second convolution incrementally. We notice the following relation if *B* is moved by *(a,b)*

$$X'(i,j) = X(i-a, j-b) \tag{8}$$

Let $T(i,j) = X(i,j) \otimes f(i,j)$, according to the definition of convolution, we have:

$$T(i-a, j-b) = X'(i, j) \otimes f(i, j) \qquad (9)$$

Equation 9 shows that if block *B* is moved by *(a, b)*, the subsequent convolution of *X'* and *f* can actually be obtained by shifting the convolution of *X* and *f* by *(a,b)*, as shown in Figure 4.11.

0	0	0	0	0
0	0	0	0	0
0	0	0	0.13	0.2
0	0	0	0.78	0.33
0	0	0	0.36	0.47

0.56	0.53	0.35	0.27	0.39
0.50	**0.49**	**0.33**	**0.25**	**0.35**
0.34	**0.34**	**0.25**	**0.20**	**0.26**
0.30	**0.30**	**0.22**	**0.18**	**0.23**
0.43	**0.41**	**0.29**	**0.23**	**0.31**

0	0	0	0	0
0	0	0	0.13	0.20
0	0	0	0.78	0.33
0	0	0	0.36	0.47
0	0	0	0	0

0.50	**0.49**	**0.33**	**0.25**	**0.35**
0.34	**0.34**	**0.25**	**0.20**	**0.26**
0.30	**0.30**	**0.22**	**0.18**	**0.23**
0.43	**0.41**	**0.29**	**0.23**	**0.31**
0.56	0.53	0.35	0.27	0.39

Figure 4.11: *X* and *X'* in Figure 5 are shown in the left column and the convolutions are in the right column. *X'* is obtained by shifting X up by 1, and the convolution of *X'* is obtained by shifting the convolution of *X* up by one.

This important property helps us to enhance the algorithm in Figure 4.9 into a fast incremental algorithm, as shown in Figure 4.12. In this fast algorithm, we first decompose the matrix into a constant term *C* and a variable term *X*, and then apply the DFT/IDFT operation to these two terms respectively to get the post-IDFT results *K* and

Y. In the loop, we only need to shift *Y*, update the sum of *K* and *Y*, and keep track of the optimal result.

```
input: a MxN matrix D, a pxq sub-matrix B, a range
([x, x+p+s-1], [y, y+q+t-1], the low pass filter
function f(i,j).
output: the optimal position of B stored in BestPostion
F = DFT(f); //discrete Fourier Transformation
BestTP = infinity;
BestPosition = (x,y)
decompose D into C and X such that D = C + X.
K = IDFT(DFT(C).F); // . represents dot product
Y = IDFT(DFT(X).F);
for (i=0; i<=s; i++)
      for (j=0; j<=t; j++)
            Shift Y by (i,j);
            rho = K+Y;
            TP = max {rho} - min {rho}
            if TP < BestTP
                 {
                  BestTP = TP;
                  BestPosition = (x+i,y+j);
                 }
```

Figure 4.12: The fast SBPP algorithm

In the algorithm, we can see both *DFT* and *IDFT* operations that are in the major loop of the simple algorithm are replaced by a simple matrix-shifting operation that is only linear time. The complexity reduces to *O(nlogn+stn)*.

This algorithm can be further improved with the expense of extra storage space. Specifically, the time of the matrix shifting operation can be reduced by using more

memory, a "space for time" technique. We notice the matrix *Y* is shifted in each iteration step. If we duplicate *Y* into a *2M*x*2N* matrix and map the index of the matrix element properly, the data movement in the matrix-shifting operation can be saved by array index remapping. Considering the low cost of memory storage nowadays, the speedup is achieved with little expense.

Figure 4.13 demonstrates the array index remapping technique. As we can see from the figure, if we want to right shift the 2x2 matrix *Y* by 1, we only remap the starting and ending indices of *Y* into the 2x2 sub-matrix marked by bold font in the extended 4x4 matrix Y^*. For the left, up and down shifts, the similar operation can be performed.

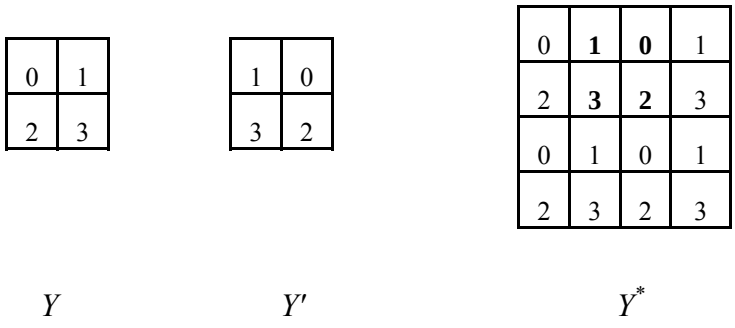


Figure 4.13: The array index remapping technique saving the data movement of *Y*.

The algorithm can also be easily generalized to solve the multi-block positioning problem that occurs in partition-based shuttle mask floorplanning problems. We only need to decompose the matrix into a set of matrices as we did in SBPP. Then we shift each non-constant matrix respectively using the incremental operation, and then sum up

these matrices to get the estimation of post-CMP topography variation, rather than apply FFT to the whole matrix.

4.4.4 Experimental Results

We implement the simple SBPP algorithm in Figure 4.9 and the fast SBPP algorithm in Figure 4.13 with extra memory storage used to further improve the efficiency. The code is written in C. For fair comparison, in the simple algorithm we use FFTW3.0.1 to computer Fourier transformation and its inverse operation. The performance of the FFTW package "is typically superior to that of other publicly available FFT software, and is even competitive with vendor-tuned codes"[72].

The code is executed on a Xeon Linux workstation. The hardware configuration is dual hyper-thread Xeon 3.4 G Hz CPU, with 1M L2 cache and 2G DRAM. We use gcc-3.3.2 to compile the code.

We test three data sets with different sizes: 300x200, 400x300, and 600x400. The feature density distribution is generated randomly. The size of the matrix is in the same order of magnitude as the one used in industry. For the 300x200 data set, we set the size of B to be 30x30. For 400x300 and 600x400 data sets, we set B to be 60x40. Each data set has four (s,t) configure: (10,5), (20,5), (20,10), and (30,10).

Table 4.2 shows the comparison between the simple SBPP algorithm and the fast algorithm. The first column refers to the test case. The tuple (300,200,10,5) represents the case that D is 300x200 (and B is 30x30), and s, t are 10 and 5 respectively. The second and the third column refer to the length of the run time for the simple algorithm and our fast algorithm respectively. The fourth column shows the speedup. As we see from the

table, our fast algorithm can obtain 6x to 9x speedup consistently. In addition, the speedup increases as the scale of the problem goes up.

(*M, N, s, t*)	Simple Algo	Fast Algo	Speedup
(300,200,10,5)	0.68s	0.11s	**6.2x**
(300,200,20,5)	1.36s	0.18s	**7.2x**
(300,200,20,10)	2.70s	0.34s	**7.9x**
(300,200,30,10)	4.08s	0.48s	**8.5x**
(400,300,10,5)	1.50s	0.23s	**6.5x**
(400,300,20,5)	3.01s	0.37s	**8.1x**
(400,300,20,10)	6.01s	0.68s	**8.8x**
(400,300,30,10)	9.02s	0.99s	**9.1x**
(600,400,10,5)	2.93s	0.44s	**6.7x**
(600,400,20,5)	5.86s	0.73s	**8.0x**
(600,400,20,10)	11.70s	1.33s	**8.8x**
(600,400,30,10)	17.48s	1.90s	**9.2x**

Table 4.2: Comparison of run-time between the simple and the fast SBPP algorithm

4.5 A NOVEL CMP DUMMY FILL PROBLEM FOR REDUCTION OF IMAGE DISTORTION

In Section 4.3 and Section 4.4, we present studies on optimization of the post-CMP topography variation by applying the low pass filter model and utilizing the LP formulation on optimal scheme of CMP dummy fill introduced in Section 4.2. In this section, employing the same model, we look at the optimality of CMP dummy fill from another perspective.

4.5.1 A Closer Look at the Measurement of Planarity

The fundamental purpose of CMP process, as we have discussed in Section 4.1, is to achieve the planarity of the wafer surface. The ideal case, obviously, is a perfectly flat wafer surface. However, when topography variation exists, the measurement of the variation concerns us, and it motivates us to take a closer look at this issue.

In the previous optimization problem of CMP dummy fill problem, the measurement of the variation is defined as the difference in height between the peak and the valley on the wafer surface, or the height spread. The smaller the spread, the better the planarity of the wafer surface. When the low pass filter model is applied, the problem is naturally formulated as a linear programming problem, as the spread takes a linear form and the amount of dummy fill to be added in any tile follows linear constraints as well.

The height spread is a good measurement of the planarity because decreasing height spread of the wafer surface can result in an increasing tolerance of lithography defocus. In this context, the defocus is defined as the distance from the best lithography focus to the wafer position [74].

To prove this, consider a perfectly flat oxide layer. Let the current best focus of the lithography process be h. On top of this oxide layer, a metal interconnect layer will be deposited. Assume the biggest defocus the lithography process can bear is d, which means that the actual position of the perfect wafer surface is acceptable as long as it falls into $[h-d, h+d]$. Therefore, the tolerance of defocus will be $2d$.

Now consider an imperfect surface with the spread e. Obviously, the focus of the peak region on the wafer surface is no less than $h-d$, and the focus of the valley region is

no more than $h+d$. Otherwise, either the peak region or the valley region falls out of the acceptable focus. For any region R on the wafer surface, assume its difference in height from the peak to be p and from the valley to be v, $p+v=e$. To ensure that this region is within the acceptable focus range, its focus must be no more than $h-d+p$ and no less than $h+d-v$. Therefore, the tolerance is $2d-e$ for any region on the wafer surface, as shown in Figure 4.14. Obviously, the tolerance is a decreasing function of the spread. When e is zero, we have the ideal case of maximum tolerance.

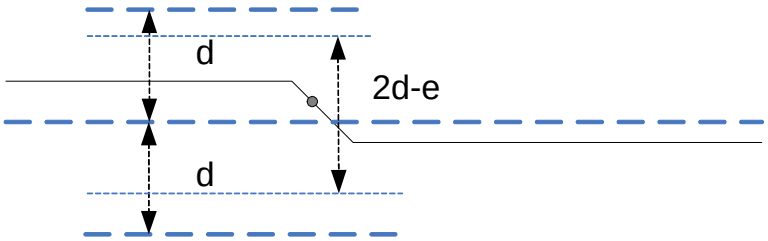


Figure 4.14: The defocus tolerance for wafer surface with topography variation spread e. The bold dash lines, from the top to the bottom, represent the plane with focus $h-d$, h, and $h+d$ respectively. The dark dot represents arbitrary region on the wafer surface. The defocus must be within the thin dash lines in order to ensure all regions are with in the acceptable focus range, $[h-d, h+d]$.

4.5.2 The Estimation Function for Image Distortion by Defocus

Although the spread measures the defocus tolerance, it does not provide guidance to where the wafer surface should be located in order to get the less image distortion, which is the ultimate goal of the lithography process improvement. To address this issue, we consider the ideal case of a perfectly flat surface again. Obviously, the best position is the in-focus one, i.e., the one with the best focus h. However, when the topography

variation exists, there are always regions that cannot be put at the in-focus position. In this situation, we need to find out an estimation function to measure and optimize the image distortion by defocus.

Given a layout X and defocus d. If we only consider the defocus, the lithography process is a function f with variables X and d, such that when d is zero, $f(X,d)=X$, which implies that the lithography has perfect image fidelity when defocus is 0. We further define $e(X,d)$ as the image distortion function: $e(X,d)=F(X,d)-X$. Function $e(X,d)$ should possess the following properties:

1. $e(X,d) = e(X,-d)$, which means the defocus has a symmetric effect on the image distortion.

2. $e(X,a) < e(X,b)$ if $0<a<b$, i.e. $e(X,d)$ is monotonically increasing in d when d is non-negative. This means that the bigger the defocus, the bigger the image distortion.

3. $e(X,0)=0$, i.e., if the defocus is eliminated, the distortion will disappear as well.

The analytical form of $e(X,d)$ is very complicated. However, given a layout X, we can use the square function as the first order approximation: $e(X,d)=d^2$ [75, 76]. If we discretize the wafer surface and consider each tile, we have the estimation function for the image distortion of the whole layout as:

$$E(X) = \sum w_i e(t_i, d_i) = \sum w_i d_i^2$$

where:

t_i refers to the sub-layout within tile i,

d_i is the defocus on tile i.

X is the whole layout.

w_i reflects the sensitivity of the layout t_i.

4.5.3 Minimization of Image Distortion by Defocus

With the estimation function derived in Section 4.5.2, we can search for the best focus for the wafer surface with topography variation. Given a layout X that consists of n tiles t_i. Each tile has a height h_i. Let D be the optimal focus for the whole layout. Obviously, the defocus for each tile d_i is equal to $(D–h_i)$. Furthermore, let $E(X) = \sum w_i d_i^2 = \sum w_i (D - h_i)^2$. The problem of minimizing image distortion by defocus is formulated as follows:

$$\min \sum w_i (D - h_i)^2$$

subject to: $h_{\min} \le D \le h_{\max}$

In the simple case w_i is assumed to be identical, let x be D. This is a single-variable quadratic programming problem [77]. We have:

$$\begin{aligned} &\sum (x - h_i)^2 \\ &= nx^2 - 2\sum h_i x + \sum h_i^2 \\ &= n(x^2 - 2x\sum h_i / n + \sum h_i^2) \end{aligned}$$

It is easy to get the optimal solution: $x = \sum h_i / n$, which is exactly the mean of h_i, and the optimal objective value is proportional to the focus variance $n\sigma^2$, where σ stands for the standard deviation of h_i. The results show that given a wafer surface with topography variation, the optimal focus in terms of minimizing the defocus effect is at the arithmetic average of the layout heights, and the estimation of minimum

image distortion is the focus variance, which answers the question raised at the beginning of Section 4.5.2.

4.5.4 The Novel CMP Dummy Fill Problem

In the single-variable of quadratic programming problem in Section 4.5.3, when we consider adding CMP dummy fill into the layout to improve the topography variation, h_i becomes a variable, instead of a constant. Consequently, the estimation of the image distortion will also be a function of dummy fill. A new optimization problem naturally arises.

We propose such a new CMP dummy fill problem that seeks to minimize the estimation of image distortion by defocus, i.e., the variance of h_i,. Based on the low pass model introduced in Section 4.2, we have the following formulation:

Minimize $\sigma^2(\rho_i)$

subject to: $0 \le l_i \le d_i \le u_i \le 1$

$\rho_{\max} - \rho_{\min} \le \varepsilon$

where:

ρ_i is the oxide pattern density of tile t_i, which is proportional to the height h_i.

l_i and u_i are the lower bound and the upper bound of the capacity of tile t_i to contain dummy fill.

d_i is the feature density in tile t_i.

$\rho_{\max}$ and $\rho_{\min}$ are two auxiliary variables that control the spread.

According to [65], the oxide pattern density is a linear combination of feature density d_i. We introduce the last constraint to guarantee that we will not sacrifice the

variation too much, as the defocus tolerance is still important. This is a typical quadratic programming problem and can be solved optimally [77]. Once the optimal solution is found, we take the arithmetic average of the height as the best focus to achieve the minimum image distortion.

The differences between our new CMP dummy fill problem and the traditional one is as follows. First, the traditional problem aims at increasing the defocus tolerance while our new problem targets at finding the optimal global focus and reducing the image distortion. Second, our problem is formulated as a QP problem while the traditional one is a LP problem. Third, in most cases, these two problems will lead to different solutions and it is the manufacturer's choice to take one of these two alternatives in different situations.

Given a design layout, the complete flow to formulate and solve the new CMP dummy fill problem is summarized as follows:

(1) Discretize the layout into small tiles.

(2) Measure the density and capacity of each tile. Here, density refers to the area of original features within a tile divided by the tile area, and capacity refers to the maximum amount of dummy fill that can be added into this tile. Next, the process parameters needs to be determined. At the end of this step, the specific objective and constraints are decided.

(3) Generate the script of the quadratic programming problem formulation for the optimization solver. As the objective and constraints are complicated and error-prone when the problem scale is large, it is preferred to automatically generate the script.

(4) Run optimization solver to get the optimal solution.

4.5.5 Experimental Results

We implement the flow of formulating and solving the new CMP dummy fill problem in 4.5.4, and compare the results of CMP dummy fill scheme with the result obtained by the traditional LP problem. We write a generator of the CPLEX linear and quadratic programming descriptions. The code is written in C. The code runs on a Xeon Linux workstation. The hardware configuration is dual hyper-thread Xeon 3.4 G Hz CPU with 1M L2 cache and 2G DRAM. We use gcc-3.3.2 to compile the code. We use CPLEX[73] in the fourth step of the flow to solve the problem.

We use the randomly generated density and capacity. Of course, we put the constraint that the sum of the density and the capacity within one tile cannot be great than 1. In addition, the process parameter and the layout size are derived from the real industry circuit.

For comparison, we run both the traditional LP and the new QP with the same test data set. Table 4.3 shows the spread and the variance comparisons between LP and QP. We use LP as the benchmark, i.e., the spread and the variance of LP are both normalized as 100%. The results show that QP does have a significant variance reduction by 45%, while the spread is increased by 23% as well. Figure 4.15 shows the topography variation of the surface after the dummy feature obtained by QP is added.

The problem	Spread	Variance
LP	100%	100%
QP	123%	55%

Table 4.3: The comparison of spreads and variances obtained by LP and QP respectively.

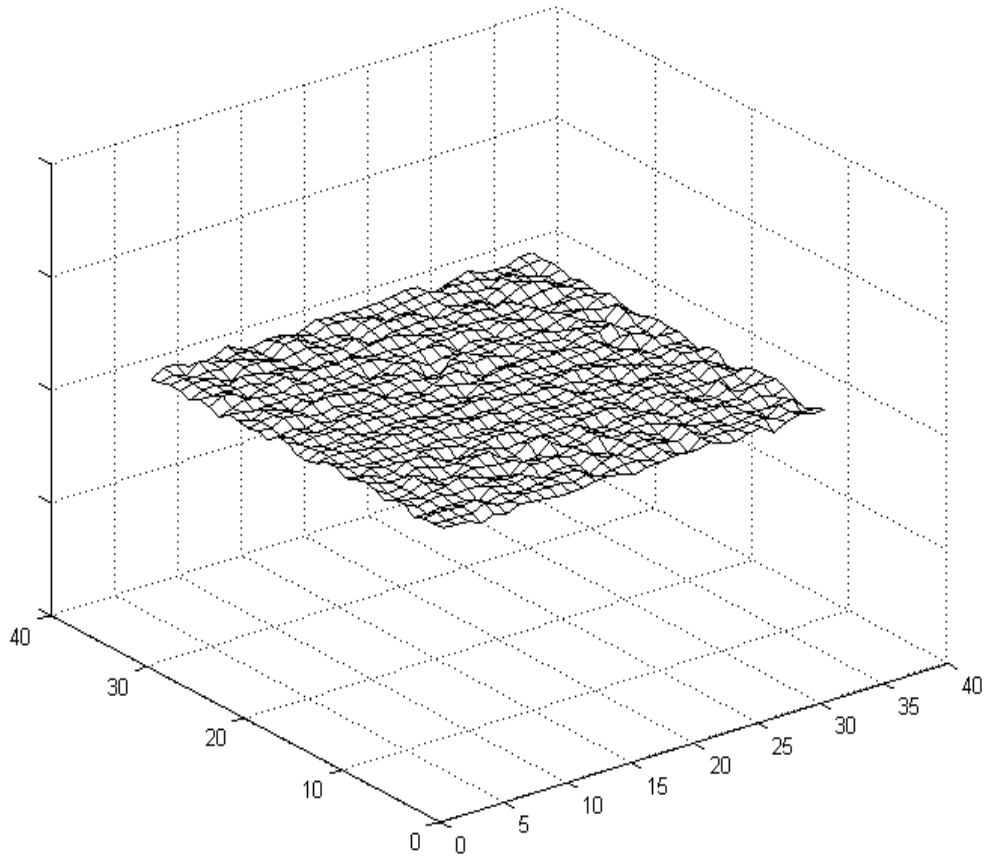


Figure 4.15: The topography variation after dummy fill obtained by QP is inserted.

4.6 CONCLUSION

In this chapter, based on a low pass filter model for post-CMP topography variation, we present several studies on optimization of the variation, include a CMP aware shuttle mask floorplanner, a fast incremental algorithm for the calculation of post-CMP variation, and a novel CMP dummy fill problem that targets for minimizing the the image distortion. The shuttle mask floorplanner along with the fast incremental algorithm

can help quickly design a shuttle mask such that the spread of the post-CMP topography variation can be minimized. The new CMP dummy fill problem, on the other hand, can reduce the image distortion by defocus as the experimental results show.

Chapter 5: Conclusions and Discussion

Integrated circuit manufacturing has become a grand challenge ever since the feature size of the transistor reaches the nanometer scale. The horn is blown to call for cooperation between both manufacturer and designer to conquer the challenge. In this dissertation, we present several layout optimization algorithms to answer the call, including a redundant via enhanced maze routing algorithm for yield improvement, a shuttle mask floorplanner, and optimization of post-CMP topography variation. These algorithm are proposed from not only manufacturing but also design perspectives to address problems in IC manufacturing.

We first present a redundant via enhanced maze routing algorithm for yield improvement. Different from the previous work that conducts redundant via insertion after routing is done, our work is the first one to add redundant via in the detail routing stage. Experimental results show that the algorithm can achieve remarkably higher rate of redundant via insertion.

Our routing algorithm is formulated as a constrained shortest path problem. Specifically, we constrain the number of dead vias, i.e., vias that cannot have redundant via, in each net. When the constraint is strong, the algorithm usually needs more time to get the solution. When run time of the router is a concern, two possible strategies can be applied to accelerate the algorithm in the future work:

(1) The constraint can be relaxed according to the sensitivity of the net with respect to dead vias. For those non-critical nets, for example, the nets not on the critical path, we can allow them to have more dead vias. A loose constraint usually leads to a fast search for the solution.

(2) The number of dead vias can be added as a penalty term into the primary objective of detail routing, wire length. In this case, although we cannot guarantee the number of dead via for each net any more, the routing problem becomes a conventional shortest path problem, and can be solved optimally in polynomial time. In addition, the penalty term will still guide the router to avoid a route with too many dead vias.

The second topic we studied in this dissertation is a shuttle mask floorplanner. Our shuttle mask floorplanner is a simulated annealing based floorplanner using slicing tree as the topological representation of the floorplan. Objectives such as area minimization and wafer utilization, and constraints such as die-to-die inspection constraint, are addressed in this work.

More interesting work can be done along this direction. For example, when there are more chips trying to get on the shuttle, the total area may close to the maximum size of the mask. In this case, the floorplanning problem could become a fixed-outline floorplanning problem[78]. Furthermore, if there are even more chips such that the total area is more than the maximum area of one shuttle mask, a shuttle mask assignment problem can be developed: how to optimally assign these chips into multiple shuttles? Of course, CMP aware shuttle mask floorplanning in Chapter 4 is also a new extension of Chapter 3.

Finally, in Chapter 4 we present several studies on optimization of post-CMP topography variation. In the first study, we propose a 3-step procedure to find the optimal shuttle mask floorplan with respect to post-CMP topography variation. In addition, we propose three predictive functions to guide the simulated annealing search. In the second study, we develop a fast incremental algorithm to quickly determine the optimal position of a chip within its enclosing rectangle in the partition-based shuttle mask floorplan. As the matter of fact, this work could be coupled into the CMP aware shuttle mask

floorplanning to enhance the second step of sliding. In the last work on post-CMP variation, we analyze the traditional linear spread objective of CMP dummy fill problem and propose a novel problem formulation with the objective of minimizing the variance of the wafer surface height so as to reduce the image distortion by defocus.

The works in Chapter 4 are based on a low pass filter model that is for oxide CMP and STI CMP. Copper CMP becomes more popular and important as more and more ASIC designs are moving to the 130nm technology node and below. Whether and how these studies in Chapter 4 can be accommodated in the case of copper CMP remain not only interesting but also challenging questions.

As we emphasize in this dissertation, IC manufacturing becomes more difficult and challenging when the feature size continuous to shrink. The research work presented in this dissertation just reveals the tip of an iceberg. Wherever there are challenges, there are opportunities. A lot more problems are waiting for scholars and industry engineers to solve Integrated circuit manufacturing Integrated circuit manufacturing, not only effectively and efficiently, but also concisely and elegantly.

Bibliology

1. Jack Kilby. "Turning Potential into Reality: The Invention of the Integrated Circuit". Nobel Lecture, 2000.

2. International Technology Roadmap for Semiconductors, 2001.

3. Gordon Moore. "Keynote Speech". IEEE International Electronic Devices Meeting, 1996.

4. F.M. Schellenberg. "Design for manufacturing in the semiconductor industry: the litho/design workshop". International Conference on VLSI Design, page 111-119, January 1999.

5. Chris Mack. "Why is semiconductor lithography hard"(manuscript).

6. "Sub-wavelength gap challenge". http://www.synopsys.com.

7. Chris Mack, "Resolution enhancement technologies". Microlithography World, May 2003.

8. Chong-Cheng Fu, Tungshen Yang, and Douglas Stone. "Enhancement of lithography patterns by using serif features". IEEE Transactions on Electron Devices, 38(12): 2599-2603, December 1991.

9. K. Harazaki, Y. Hasegawa, Y. Schichijo, H. Tabuchi, and K. Fujii. "High accurate optical proximity correction under the influences of lens aberration in 0.15um logic process". International Microprocesses and Nanotechnology Conference, page 14-15, 2000.

10. K. Yamamoto, S. Kobayashi, T. Uno, T. Kotani, S. Tanaka, S. Inoue, S. Watanabe, and H. Higurashi. "Hierarchical optical proximity correction on contac hole layers". International Microprocesses and Nanotechnology Conference, page 40-41, 2000.

11. N. Cobb and A. Zakhor. "Large area phase-shif mask design". Proceedings of SPIE, volume 2197, page 348-359, 1994.

12. N. Cobb, A. Zakhor, and E. Miloslavsky. "Mathematical and CAD framework for proximity correction". SPIE, volume 2726, page 208-222, 1996.

13. Y.C. Pati, Y.T. Wang, J.W. Liang, and T. Kailiath. "Phase shift masks: automated design and mask requirement". SPIE, volume 2197, page 314-327, 1994.

14. Marc D. Levenson, N.S. Viswanathan, and Robert A. Simpson. "Improving resolution in photolithography with a phase-shift mask". IEEE Transactions on Electron Devices, 29(12):1828-1836, December 1982.

15. Chris Mack, "Off-axis illumination" Microlithography World, August 2003.

16. Li-Da Huang and Martin D.F. Wong. "OPC-friendly maze routing". Design Automation Conference, 2004.

17. Brion Technologies. TACHYON Computational lithography system.

18. Alfred J. Reich, Kent Nakagawa, and Robert Boone. "OASIS versus GDSII stream format efficiency". SPIE, volume 5256, 2003.

19. Gang Xu, Ruiqi Tian, Martin D.F. Wong, and Alfred Reich. "Shuttle mask floorplanning". SPIE, volume 5256, 2003.

20. L. Capodieci, P.Gupta, A. Kahng, D. Sylvester, and J. Yang. "Toward a methodology for manufacturability-driven design rule exploration". Design Automation Conference, page 331-316, 2004.

21. TSMC Symposium at Austin. 2004.

22. P. Gupta and A.B. Kahng. "Manufacturing-Aware Physical Design". International Conference on Computer-aided Design, page 681-687, 2003.

23. Louis K. Scheffer. "Physical CAD Changes to Incorporate Design for Lithography and Manufacturability". Asia and South Pacific Design Automation Conference, 2004.

24. Y. Zorian, D. Gizopoulos, C. Vandenberg, and P. Magarshack. "Guest Editors' Introduction: Design for Yield and Reliability". IEEE Transaction on Design and Test of Computers, volume 21, issue 3, May 2004.

25. MOSIS Test Results for TSMC 0.18um Run. http://www.mosis.org

26. S. Wolf. "Introduction to Dual-Damascene Interconnect Process". Silicon Processing for the VLSI Era, volume 4, pp 674-679, Lattice Press, 2004.

27. J. Lienig and G. Jerke. "Electromigration-Aware Physical Design of Integrated Circuits". VLSI Design Conference, 2005.

28. H. Okabayashi. "Stress-induced Void Formation in Metallization for Integrated Circuits". Material Science Engineer, R11, page191-241, 1993.

29. T. D. Bonifield, J. C. Ondrusek, and W. R. McKee. "Stress-Induced Voiding Under Vias Connected To Wide Cu Metal Leads". IRPS, 2002.

30. P. Gelsinger. Keynote. Design Automation Conference, 2004.

31. B.Y. Su and Y.W. Chang. "An Exact Jumper Insertion Algorithm for Antenna Effect Voidance/Fixing". Design Automation Conference, page 325-328, 2005.

32. J. Kibarian. "Ramping New Products Yields Ramping New Products Yields in the Deep in the Deep Submicron Age". International Symposium on Quality Electronic Device, 2000.

33. G. A. Allan and A. J. Walto. "Automated Redundant Via Placement for Increased Yield and Reliability". SPIE, volume 3216, page 114-125, 1997.

34. Hardy K.S. Leung, "Advanced Routing in Changing Technology Landscape", International Symposium on Physical Design, page 118-121, 2003.

35. Naveed Sherwani. "Algorithms for VLSI Physical Design Automation", 3rd edition. Kluwer Academic Publisher, 1999.

36. Hai Zhou and Martin D.F. Wong, "Crosstalk-Constrained Maze Routing Based on Lagrangian Relaxation", International Conference on Computer Design, page 628-633, 1997.

37. Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. "Introduction to Algorithms". MIT Press, 2001.

38. R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, "Network Flows: Theory, Algorithms, and Applications". Prentice Hall, 1993.

39. S. Chen and E. C. Lynn. "Effective placement of chips on a shuttle mask". SPIE, volume 5130, page 681-688, 2003.

40. G. Xu, R. Tian and D. F. Wong, and A. Reich. "CMP-aware Shuttle Mask Floorplanning". Asian and South Pacific Design Automation Conference, 2005.

41. G. Xu, R. Tian, and D. F. Wong. "A multi-objective floorplanner for shuttle mask optimization". SPIE, volume 5567, 2004.

42. M. Andersson, J. Gudmundsson, and C. Levcopoulos. "Chips on wafer". Workshop on Algorithms and Data Structures, 2003.

43. A.B. Kahng, I. I. Mandoiu, Q. Wang, X. Xu, and A. Zelikovsky. "Multi-project reticle floorplanning and wafer dicing". International Symposium on Physical Design, 2004.

44. A.B. Kahng and S. Reda. "Reticle Floorplanning With Guaranteed Yield for Multi-Projects Wafers". International Conference on Computer Design, 2004.

45. L. Stockmeyer. "Optimal orientation of cells in slicing floorplan design". Information and Control, volume 57, page 91-101, 1983.

46. Martin D.F. Wong and C.L. Liu, "A new algorithm for floorplan design", Design Automation Conference, page 101-107, 1986.

47. H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence pair", IEEE Transaction on Computer-aided Design, 1996.

48. Xiaoping Tang and D. F. Wong. "FAST-SP: a fast algorithm for block placement based on sequence pair". Asia and South Pacific Design Automation Conference, 2001.

49. Parquet, http://vlsicad.eecs.umich.edu/BK/parquet/

50. B. Yao, H. Chen, C.K. Cheng, and R. Graham "Revisiting floorplan representations". International Symposium on Physical Design, page 138-143, 2001.

51. Photomask Basics, www.photronics.com

52. Garey, M. and D. Johnson. "Computers and Intractability; A Guide to the Theory of NP-Completeness". W.H. Freeman, 1979.

53. D.J.A. Welsh and M.B. Powell. "An upper bound for the chromatic number of a graph and its application to timetabling problems". Computer Journal, volume 10, page 85-86, 1967-68.

54. "Tech brief: Chemical Mechanical Planarization". www.icnowledge.com.

55. I. Ali, S. Roy, and G. Shinn. "Chemical-Mechanical Polishing of Interlayer Dielectric: A Review". Solid State Technology, volume 37, page. 63-70, October, 1994.

56. A. Chatterjee et al. "A Shallow Trench Isolation Study for 0.25/0.18 μm CMOS Technologies and Beyond". Symposium on VLSI Technology, pages 156–157, 1996.

57. J. M. Steigerwald, S. P. Murarka, and R. J. Gutmann. "Chemical Mechanical Planarization of Microelectronic Materials", John Wiley & Sons, 1997.

58. D. Ouma. "Modeling of Chemical-Mechanical Polishing for Dielectric Planarization". Ph. D. Dissertation, MIT, 1998.

59. B. Lee. "Modeling of Chemical Mechanical Polishing for Shallow Trench Isolation". PhD dissertation, MIT, 2002.

60. Stine, B. E., Ouma, D. O., Divecha, R. R., Boning, D. S., Chung, J. E., Hetherington, D. L., Harwood, C. R., Nakagawa, O. S., and Oh, S.-Y. "Rapid characterization

and modeling of pattern-dependent variation in chemical-mechanical polishing". IEEE Transaction on Semiconductor Manufacturing, volume 11, page 129–140, 1998.

61. Ruiqi Tian. "Layout Optimization with Dummy Features for Chemical-Mechanical Polishing Manufacturability". Ph.D. dissertation, UT-Austin, 2002.

62. R. Tian, D. F. Wong, and R. Boone. "Model-Based Dummy Feature Placement for Oxide Chemical-Mechanical Polishing Manufacturability". Design Automation Conference, page 667-670, 2000.

63. R. Tian, X. Tang and D.F. Wong. "Filling and slotting for process uniformity control in copper chemical-mechanical polishing". In Proc. 6th international Chemical-Mechanical Planarization for ULSI Multilevel Interconnection Conference, page 57-62, March 2001.

64. R. Tian, X. Tang and D.F. Wong. "Dummy Feature Placement for Chemical-Mechanical Polishing Uniformity in a Shallow Trench Isolation Process". International Symposium on Physical Design, page 118-123, April 2001.

65. R. Tian, D.F.Wong, and R. Boone. "Model-based Dummy Feature Placement for Oxide Chemical-Mechanical Polishing Manufacturability" IEEE Transaction on Computer-aided Design, volume 20, page 902-910, 2001.

66. R. Tian, X. Tang, and D.F. Wong. "Dummy feature placement for chemical-mechanical polishing uniformity in a shallow trench isolation process". IEEE Transactions on Computer-aided Design, volume 21, issue 1, page 63-71, 2002.

67. A. B. Kahng, G. Robins, A. Singh and A. Zelikovsky. "Filling Algorithms and Analyses for Layout Density Control". IEEE Transactions on Computer-aided Design, 18(4):445–462, 1999.

68. Y. Chen, A. B. Kahng, G. Robins and A. Zelikovsky. "Area Fill Synthesis for Uniform Layout Density". IEEE Transactions on Computer-aided Design, 21(10):page 1132–1147, 2002.

69. P. Beckage, T. Brown, R. Tian, E. Travis, A. Phillips and C. Thomas. "Prediction and Characterization of STI CMP Within-Die Thickness Variation on 90nm Technology". CMP-MIC Conference, pages 267–274, 2004.

70. C. S. Burrus and T. W. Parks. DFT/FFT and Convolution Algorithms: Theory and Implementation, John Wiley and Son, 1985.

71. P. Beckage, T. Brown, R. Tian, E. Travis, A. Phillips and C. Thomas. "Implementation of Model-Based Tiling at STI CMP for 90nm Technology". CMP-MIC Conference, pages 157–162, 2004.

72. FFTW. http://www.fftw.org/

73. CPLEX. http://www.ilog.com/products/cplex/

74. Chris Mack. "Lithography Glossary". www.kla-tencor.com

75. Y. Peng, D. Pan, and C. Mack. "Fast Lithography Simulation under Focus Variations for OPC and Layout Optimizations". SPIE, volume 6156, 2006

76. Yu Peng. Personal communication.

77. D.P. Bertsekas. Nonlinear Programming, 2nd Edition. Athena Scientific, 2000.

78. S. Adya and I. Markov. "Fixed-outline Floorplanning : Enabling Hierarchical Design". IEEE Transaction on VLSI, 11(6):1120-1135, December 2003

Vita

Gang Xu was born in Xian, China on Nov 28, 1974, the younger son of Guohua Xu and Chuzhen Wang. He received his B.S. and M.S. degree in Computer Science from Peking University in 1997 and 2000, respectively. In the Fall of 2000, he entered the Graduate School of The University of Texas at Austin to pursue his Ph.D. degree. From summer 2006 to spring 2007, he worked in a startup company named Clear Shape Technologies as a Senior R&D Engineer. He published 4 papers during his graduate study in the area of CAD algorithms for VLSI design and manufacturing.

Permanent address: 1270 Coronado Drive, Apt 22, Sunnyvale, CA 94086

This dissertation was typed by Gang Xu.